2011

Secure location-aware communications in energy-constrained wireless networks

Yawen Wei
Iowa State University

Secure location-aware communications in energy-constrained wireless networks

by

Yawen Wei

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Engineering

Program of Study Committee:
Yong Guan, Major Professor
Manimaran Govindarasu
Douglas W Jacobson
Daji Qiao
Arun K Somani
Johnny S Wong

Iowa State University

Ames, Iowa

2011

DEDICATION

I would like to dedicate this dissertation to my husband Peifeng, my parents Quanshun and Maojun, and my parents-in-law Guoyong and Liangying. Without their endless love, understanding and support, I would not have been able to complete this work.

And, to my lovely daughter, Angel.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting the research and in my writing of this thesis.

First and foremost, I would like to express the sincerest gratitude to my advertiser, Dr. Yong Guan. Without his insights and knowledge, and his persistent guidance and support, this dissertation would not have been possible. I thank him for the countless hours he has spent with me discussing my research and revising my papers. I will never forget his encouragements that renewed my hopes from time to time for overcoming difficulties and completing my graduation studies. His breadth of knowledge, rigorous scholarship, and persistence of pursuing high standards will always inspire me and have a profound influence on my future career and life.

I would also like to thank my committee members: Dr. Daji Qiao, Dr. Douglas W Jacobson, Dr. Manimaran Govindarasu, Dr. Johnny S Wong, Dr. Akhilesh Tyagi, and Dr. Arun K Somani. Their valuable advices and assistance helped me to improve the quality of my thesis. I am also thankful to Dr. George Amariucai for his valuable discussions.

I am greatly thankful to my labmates: Zhen Yu, Linfeng Zhang, Lars Kulseng, Yanlin Peng, Chunwang Gao, Yang Liu, Wenji Chen, Gang Xu, and Ramzi Saifan, for their collaborations during this research. I appreciate all my friends at Iowa State University for providing me a wonderful life in Ames, which will be the most beautiful town in my memory.

Finally, I would like to convey special thanks to my husband Peifeng Zhang, my parents Quanshun Wei and Maojun Xing, and my parents-in-law Guoyong Zhang and Liangying Zhou, for their love, understanding and support in my life.

ABSTRACT

Wireless ad hoc network has enabled a variety of exciting civilian, industrial and military applications over the past few years. Among the many types of wireless ad hoc networks, Wireless Sensor Networks (WSNs) has gained popularity because of the technology development for manufacturing low-cost, low-power, multi-functional motes. Compared with traditional wireless network, location-aware communication is a very common communication pattern and is required by many applications in WSNs. For instance, in the geographical routing protocol, a sensor needs to know its own and its neighbors' locations to forward a packet properly to the next hop.

The application-aware communications are vulnerable to many malicious attacks, ranging from passive eavesdropping to active spoofing, jamming, replaying, etc. Although research efforts have been devoted to secure communications in general, the properties of energy-constrained networks pose new technical challenges: First, the communicating nodes in the network are always unattended for long periods without physical maintenance, which makes their energy a premier resource. Second, the wireless devices usually have very limited hardware resources such as memory, computation capacity and communication range. Third, the number of nodes can be potentially of very high magnitude. Therefore, it is infeasible to utilize existing secure algorithms designed for conventional wireless networks, and innovative mechanisms should be designed in a way that can conserve power consumption, use inexpensive hardware and lightweight protocols, and accommodate with the scalability of the network.

In this research, we aim at constructing a secure location-aware communication system for energy-constrained wireless network, and we take wireless sensor network as a concrete research scenario. Particularly, we identify three important problems as our research targets: (1) pro-

viding correct location estimations for sensors in presence of wormhole attacks and pollution attacks, (2) detecting location anomalies according to the application-specific requirements of the verification accuracy, and (3) preventing information leakage to eavesdroppers when using network coding for multicasting location information. Our contributions of the research are as follows: First, we propose two schemes to improve the availability and accuracy of location information of nodes. Then, we study monitoring and detection techniques and propose three lightweight schemes to detect location anomalies. Finally, we propose two network coding schemes which can effectively prevent information leakage to eavesdroppers. Simulation results demonstrate the effectiveness of our schemes in enhancing security of the system. Compared to previous works, our schemes are more lightweight in terms of hardware cost, computation overhead and communication consumptions, and thus are suitable for energy-constrained wireless networks.

CHAPTER 1. OVERVIEW

1.1 Introduction

Wireless ad hoc network [78] has enabled a lot of exciting civilian, industrial and military applications, and it has been receiving intensive attention from academia and industry over the past few years. Wireless ad hoc network is generally consisted of a collection of energy-constrained (e.g., battery-powered) wireless devices, which self-configure to form a multihop network without any prearranged infrastructure such as routers or access points. The nodes communicate with each other to maintain connectivity and to handle the control tasks in a collective and distributed manner. Among the many types of wireless ad hoc networks [1, 2, 42], Wireless Sensor Networks (WSNs) has gained popularity because of the development of Micro-Electro-Mechanical System (MEMS) technology for manufacturing low-cost, low-power, multi-functional motes. WSNs have been used for a variety of real-life applications, for example, habitat monitoring where nodes monitor environmental parameters such as temperature, humidity and atmospheric pressure; traffic controlling where nodes detect moving vehicles and estimate their speed and direction; and battlefield surveillance where sensor nodes gain information such as enemy movements and explosive attacks.

As WSNs come to be wide-spread deployment, security problems arise and become a central concern. The security issues are originated because of the following two properties of WSNs: First, use of wireless links for communications renders the network susceptible to link attacks ranging from passive eavesdropping to active spoofing, jamming, replaying, etc. Passive adversaries with appropriately designed antenna and receiver can easily pick off the radio transmissions to gain confidential information. Active adversaries can interfere with the radio frequencies in the network to disrupt the transmitted information. They can also delete or

modify messages, inject erroneous messages or replay previously heard messages. Second, since effective tamper-resistance hardware adds significant cost to the network, WSNs are mostly unguarded. Adversaries can damage or replace sensor node, capture a node and extract cryptography materials stored on it, or deploy multiple malicious nodes that collude together to attack the system. They can also launch sybil attacks [23], wormhole attack [50], flooding attack [44], sinkhole attack [81] and much more. All these attacks disrupt the operation of WSNs and violate basic requirements for secure communications.

Secure communication is very important for WSNs, especially for some sensitive applications such as burglar alarms, military surveillance, and emergency response. Although much research effort has been devoted to secure wireless communications in general, the properties of WSNs pose new technical challenges: First, sensor nodes are designed to be unattended for long periods without any physical maintenance (e.g., battery recharging/replacement), so that their energy resource is a premier. Second, sensor nodes have limited local memory, computation capacity, communication range and bandwidth. Third, the number of sensor nodes in WSNs can be potentially several orders of magnitude higher than that in general wireless ad hoc network. These properties make it infeasible to utilize existing secure algorithms designed for conventional wireless networks. Therefore, innovative mechanisms should be designed that can conserve power consumption, use inexpensive hardware and lightweight protocols, and accommodate with the scalability aspect of WSNs.

Location-aware communication is required by many applications, and becomes a very common communication pattern in WSNs. Actually, lots of applications are designed to rely on the availability of sensors' locations to function appropriately. For instance, in battlefield surveillance, a sensor that detects a tank needs to report the location of the event to the base station. In spatial IP address assignments, each sensor needs to know its physical location to construct its IP address. In geographical routing protocol, a sensor must know its own and its neighbors' locations in order to forward the packet to the one that is closest to the destination.

In this research, we focus on wireless sensor networks and aim at designing secure location-aware communication system. Particularly, we identify three key aspects when building such

a system: secure localization, location claims verification, and distribution of the confidential location information. We investigate a variety of malicious attacks and design several practical techniques. First, we propose two schemes to enhance the availability and accuracy of sensor nodes' location estimations. Then, we study the monitoring and detection techniques and propose three lightweight schemes to detect location anomalies. Finally, we propose two network coding schemes which can effectively prevent information leakage when distributing location information in WSNs.

1.2 A Motivating Scenario

In this section, we use the battlefield surveillance application as a motivating scenario to describe some of the challenging problems when we construct a location-aware communication system for WSNs.

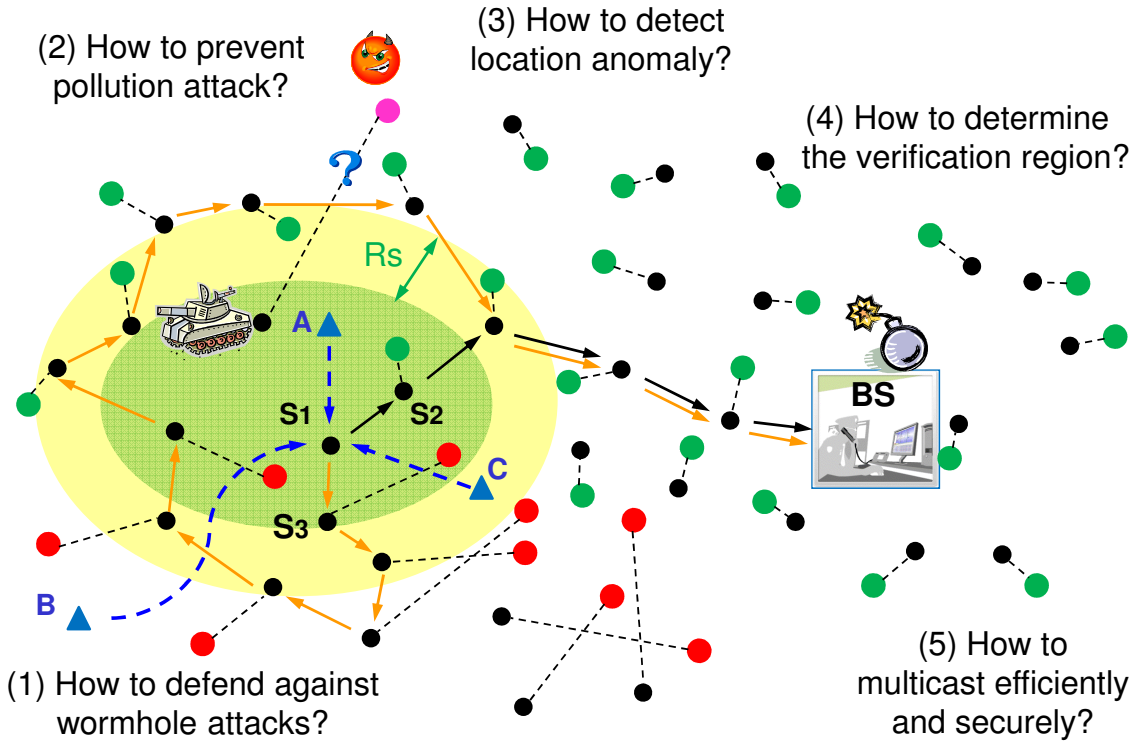


Figure 1.1 A Motivating Scenario

Battlefield surveillance is one of the many location-based applications for WSNs. As shown

in Figure 1.1, many sensor nodes are randomly deployed through a battlefield for monitoring military-related phenomenons. A base station serves as the control center, collecting reports and assigning tasks to sensors. Some special sensors are equipped with GPS device, so they can determine their locations directly. These sensors are named *anchors* (illustrated by the blue triangles) and they periodically broadcast beacon messages to normal sensors to help them localize. To disrupt the localization process, the adversaries may create a wired link, record messages at one end of the tunnel, transmit them through the tunnel and replay the messages at the other end. In the figure, such a tunnel is shown by the dashed line between anchor B and sensor S_1. As a result, sensor S_1 will mistakenly think it is close to anchor B, and estimate a wrong location far away from its true one. This attack is generally called the "wormhole attack" to the Wireless Sensor Networks. The challenge problem is how sensors can still estimate locations correctly in presence of wormhole attacks.

The battlefield is very large and it is possible that not all sensors can receive beacon messages from anchors. Therefore, sensors that determine their locations serve as location references for other sensors that are not yet localized. In the location propagation process, if a sensor estimates a wrong location and broadcasts it, many downstream sensors' localization may be impacted. As shown in the figure, many sensors at the bottom left area estimate their locations with large errors. Such attack is called "pollution attack". Location-based applications will be malfunctioned if these wrong locations are not detected. For example, if sensor S_1 detects a tank and tries to notify the base station, and if geographical routing protocol is used, S_1 should forward the packet to sensor S_2 which is the closest one among its neighbors to the base station. However, if sensor S_3 estimates a location which is closer to the base station than S_2, S_1 will probably forward the packet to S_3, which may result in a much longer route (the orange route in the figure), or the packet cannot even reach the destination. So an interesting problem we need to solve is that, considering that location propagation exists in many localization schemes, how sensors can detect wrong location reference soon and mitigate the impact of the pollution attack as much as possible.

Besides wormhole attack and pollution attack, there are many other malicious disruptions.

If sensor nodes are compromised by adversaries, then they can claim any arbitrary location to deceive the base station and other sensors. So the challenge we are facing is how to detect location anomalies and verify if a claimed location is correct in terms of localization errors. Furthermore, location anomalies are related to application functions. A location unacceptable by one application may be tolerable in another application. Therefore, it is important to determine a "verification region" specifically tuned for the applications' needs. For example, in Figure 1.1, sensor S_1 detects a tank and reports to the base station. The base station calculates the projection spot and launches a missile, which explodes a region illustrated by the yellow area in the figure. In order to determine whether the tank can be actually wrecked, it is helpful to verify whether sensor S_1 is in the green region or not, given that the distance between the tank and the sensor cannot exceed the sensing distance R_s. In this application, the green region is the "verification region" for sensor S_1. The general question is how we can determine verification region properly for each specific location-based applications and calculate the in-region probabilities of sensors.

We continue with the above application scenario where sensors report to the base station about their detection of a tank. Rather than launching the missile immediately, the base station may behave more cautiously and ask the surrounding sensors to increase their sampling rates. Basically the base station should send a packet with proper payload to several sensors. To achieve this, the base station can initiates multiple unicasts each for one destination, but this is not an efficient solution especially for energy-constrained WSNs. If multicast is used, then how can we make the multicast efficient in terms of throughput and bandwidth overhead. Network coding has been recently proposed as a promising technique to maximize network throughput and to reduce the number of retransmissions. But network coding is subjected to many attacks when used in WSNs. Thus the challenging problem we are facing now is how we can provide efficient and secure multicast in presence malicious attacks.

1.3 Objectives of Research

1.3.1 Providing Secure Localization

In this research, we first focus on how to provide correct and reliable location estimations for sensors, because many applications for WSNs are designed to rely on the ability to accurately locate each sensor in order to function appropriately. For instance, in geographical routing protocol, a sensor need to know its own and its neighbors' locations to properly forward the packet to the next hop. However, providing location information for sensor nodes is a very challenging problem. Not only the environmental disruptions are present and impossible to avoid, but also attackers can easily manipulate non-secured location information by launching various attacks.

Wormhole attack is a notorious attack, where the adversaries copy messages at one position and replay them at another location. Hence, a sensor may receive beacon messages from a far-off anchor through a wormhole tunnel, and mistakenly think the anchor is in its neighborhood. Using the wormholed reference, the sensor will estimate a wrong location or not be able to estimate a location. Since the wormholed references are generated by legitimate anchors, the integrity of these messages are not damaged. Hence, traditional security mechanisms such as encryption and authentication are not sufficient to defend against wormhole attacks. Defending against wormhole attacks has attracted much research interest in the past. These approaches, however, introduce high computation or computation overhead to sensor nodes. Since sensor nodes are limited in energy supply, memory storage and computation capacity, lightweight algorithms should be designed against wormhole attacks.

Besides wormhole attack, there are many other attacks which may cause sensors to obtain a wrong location estimation, including sybil attack, range-enlargement, and range-reduction attack, etc. If any wrongly localized sensors start to serve as reference points for other sensors, then other sensor's localization will be damaged. The proliferation of location pollution can be very fast, and the whole network's localization can be potentially impacted. Such error proliferation is called "pollution attack" which is a very detrimental attack. Therefore, we

should design secure localization schemes that can help sensor nodes to eliminate false location references as soon as possible, as a result, the pollution degree and scope can be mitigated.

1.3.2 Detecting Location Anomalies

Monitoring and detection mechanisms play an important role for achieving secure communications in a system. In case that any policy violation or unexpected behavior occurs, the system can recognize and detect anomalies and alert the administrator(s). In WSNs, detection algorithms should detect the presence of potentially hostile sensor node, and take corrective actions such as node revocations. In this research, we specifically study the problem of detecting location anomalies because location information are crucial for many applications and location anomalies can cause severe consequence. For example, when military sensor networks are used to monitor enemy movements and suspicious phenomena, a detection report with wrong location of a tank/bomb can cause significant damage.

There are lots of research efforts on designing location anomaly detection techniques, They can be classified into two categories, namely, *on-spot* verification and *in-region* verification. On-spot verification verifies whether sensors' true locations are within a certain error range from their true locations. Locations that cannot be verified are considered as anomalies. To obtain the desired on-spot verification results, existing algorithms either utilize the deployment knowledge of sensors in the field [26] or make use of some dedicated hardware to verify distance measurements [11, 15, 16, 62]. However, neither the knowledge of deployment patterns nor the required dedicated hardware is always available in WSNs, especially the general-purpose low-cost WSNs, thus, lightweight verification algorithm should be designed to overcome the shortcomings of existing algorithms and can be used widely applied to energy-constrained wireless networks.

Besides on-spot verification, in-region verification also provides proper verification results for some applications. Shankar and Wagner defined the concept of *in-region verification* [85]. They proposed a protocol named *Echo* to verify if a sensor is inside a physical region such as a room, a building, a sport stadium, etc. Based on the verification result, access right can

be properly assigned to sensors (i.e., people who take the sensors) to access some resources in that physical region. As the first work, *Echo* successfully utilizes in-region verification to facilitate location-based access, however, it cannot be directly applied in other location-based applications, because the region may not be explicit and needs to be determined carefully by analyzing applications' functions. Secondly, when performing in-region verification, *Echo* requires the use of multiple *verifiers* that can transmit radio signal and receive ultrasound signal, and bound their XOR operations within the magnitude of nanoseconds. Such verifiers increase the expense and require extra infrastructure which is not easy to construct. These limitations of *Echo* motivate us to study how to integrate application requirements into determining the physical region, and how to design verification algorithms that are lightweight enough not to depend on dedicated hardware such as *verifiers*.

1.3.3 Distributing Location Information

When distributing location information in WSNs, both unicast and multicast communication patterns can be used based on whether there are multiple destinations. While research on unicast in WSNs has produced many promising results, multicast is still an area that needs development and improvements. Recently, network coding has been proposed as a new message-forwarding technique, and it can effectively improve multicast throughput. Unlike traditional approaches where forwarders only duplicate packets, network coding allows a forwarder to process multiple input packets and encode them into a coded one.

However, network coding is vulnerable to malicious attacks. For example, if an adversary eavesdrops on the transmissions between sensor nodes, then he or she can obtain multiple coded packets and access sensitive information by solving linear/nonlinear equation systems. Traditional approaches including end-to-end encryption and hop-by-hop encryption partly solve the problem. However, end-to-end encryption are usually impractical because sensor's memory cannot afford a large number of unique encryption keys, and designing robust and efficient key distribution scheme is another challenging problem. For the hop-by-hop approach, since every pair of neighboring nodes need to share a secret key, not only extra communication overhead

will be caused by the key setup process, but also much delay will be introduced by encryption and decryption at every hop.

Besides using encryption approaches, other secure solutions for network coding have also been proposed. These solutions usually insert randomness into the network to make the messages transmitted on all links randomized. However, they consume more communication bandwidth and energy which is very constrained resource in WSNs. Therefore, we aim to design efficient and lightweight network coding schemes that can prevent information leakage, and are also suitable for energy-constrained networks.

1.4 Contributions of Research

The contributions of our research are as follows:

1. We study the problem of providing reliable location information in presence of various attacks. First, we propose a dynamic anchor-regrouping localization scheme to defend against wormhole attack. Compared with previous works, our scheme does not require any dedicated or expensive hardware on sensor nodes, such as sector antennas and high-precision measuring device. Moreover, our scheme is more energy-efficient since no extra communication for detecting wormholes are incurred on sensor nodes. Second, we proposed a robust localization scheme to mitigate the impact of pollution attack. This scheme is based on the novel notion of confidence tag which quantifies the accuracy of sensor's location estimation. To our knowledge, our scheme is the first one to address the problem of the proliferation of wrong locations. Simulation results demonstrate the effectiveness of our schemes in enhancing the availability and accuracy of location information for wireless sensor networks.

2. We also study the problem of detecting and revoking location anomalies. First, we propose two schemes to verify whether the location claimed by a sensor deviates from its true location more than a certain distance. Second, we study the in-region verification problem which is to determine whether a sensor is inside an application-specific area. The latter research takes the first step tempting to integrate the application requirements

into determining the trustability of sensors' locations. Compared to previous works, our verification algorithms do not require any verifier-infrastructure to be deployed through the sensor field.

3. For distributing location information efficiently and securely, we propose two network coding schemes which can prevent information leakage to eavesdroppers. Adversaries who eavesdrop on the wireless communications can at most obtain linear combinations of the original information symbols, yet they cannot solve for any single one of them. Compared with previous work, our advanced scheme have two nice properties: First, it does not introduce extra random information into the network and thus achieves maximum multicast capacity. Second, it does not enlarge the size of the finite field from which the coding coefficients are selected. Because the size of finite field decides the number of binary bits needed to represent coded packets, our scheme consumes less communication resource than previous works and is suitable for energy-constrained wireless networks. In addition, our network coding schemes are not limited to be used for multicasting location information, they can be used in any multicasting scenario and can preserve privacy against eavesdroppers.

1.5 Dissertation Organization

The rest of this dissertation is structured as follows: In chapter 2, we provide a review of literature for the three problems targeted in our research. In chapter 3, we study how to provide reliable location estimations for sensors and enhance the accuracy and availability of the localization. In chapter 4, we consider how to detect location anomalies and perform location verifications. In chapter 5, we research how to securely distribute location information by designing network coding schemes against eavesdropping attacks. In chapter 6, we summarize our research and discuss our future work.

CHAPTER 2. REVIEW OF LITERATURE

2.1 Secure Localization Schemes

2.1.1 Classification of Localization Approaches

In recent years, many localization approaches have been proposed for wireless sensor network. Before we talk about the security issues related to localization, we firstly take an overview of the background of localization approaches and make a classification of them.

The most traditional and widely-used localization system is the Global Positioning System (GPS). The earth-based GPS receivers can provide users with location, speed, and time, by calculating the distances from at least three satellites. However, it is not feasible to equip the relatively expensive GPS receiver on each node in large scale sensor networks, but only a fraction of them. Generally, the sensors equipped with GPS receivers (or the sensors who can obtain their locations through manual configurations) are called anchors. The current localization approaches can be classified as anchor-based or anchor-free ones on whether using anchors; range-based or range-free ones on whether using the measured distances; decentralized or centralized. We classify the localization approaches in TABLE 2.1 (where "(c)" denotes the centralized approach), and discuss them in more details.

2.1.1.1 Anchor-based range-based approaches:

Most localization approaches require that there are some anchors whose positions are already known through GPS device or manual configuration. In range-based ones, the distances between anchor and sensors and the distances between sensors can be measured, and sensors' locations are determined by trilateration, using the distances to at least *three* anchors

Table 2.1 Classification of Sensor Localization Approaches

	Range-based	**Range-free**
Anchor-based	Active Bat(c) [43] RADAR(c) [4] AHLoS [83] SDP(c) [88] LMS/KF [87]	Active Badge(c) [92] Cricket [80], Centroid [13] DV-hop [75], SeRLoc [61] Amorphous [74] DV-based AoA [76] APIT [45], Convex(c) [28]
Anchor-free	MDS-MAP(c) [89]	MDS-MAP(c) [89] Deployment Knowledge [32]

or position-known sensors. RADAR [4], Active Bat [43], AHLoS [83], and SDP [88] are all anchor-based range-based approaches. Moreover, Kalman-Filter (KF) or Least Mean Square (LMS) [87] methods can be applied together to deal with the noisy measurements of distances.

2.1.1.2 Anchor-based range-free approaches:

Since range-based approaches always require special hardware to measure the angles or distances, range-free approaches now attract more research interests. In anchor-based range-free approaches, no distance measurements are needed and sensors determine their locations using anchors' beacon messages. Active Badge [92] and Cricket [80] belong to this category. Centroid [13] simply takes the mean value of a sensor's surrounding anchors' locations as the sensor's estimated location. APIT [45] determines some triangles in which a sensor may reside, then estimates the sensor's location as the overlapping region of theses triangles. SeRLoc [61] uses sectored antennas equipped on anchors and computes sensor's location as the centroid of overlapping region of sectors. DV-hop [75] and DV-based AoA [76] obtain the hop counts to anchors by flooding through the network, estimates the average hop distance, then calculates the distances or AoA between sensors and anchors to determine sensors' locations. Amorphous [74] employs a similar strategy as DV-hop but calculates the average hop distance offline. Convex [28] formalizes the localization problem as linear equations that satisfy some connectivity-induced constraints, and utilizes Linear Programming (LP) method to obtain the optimal solutions.

2.1.1.3 Anchor-free range-based approaches:

There are relatively fewer anchor-free range-based localization approaches. One is MDS-MAP [89], which is based on multidimensional scaling technique to derive the locations of all sensors. However, it can also work as range-free approach using only the connectivity information between sensors, though with some degradations of the localization performance.

2.1.1.4 Anchor-free range-free approaches:

As we discussed, MDS-MAP [89] can work as a centralized anchor-free range-free localization approach. L. Fang, W. Du, and P. Ning proposed another decentralized approach in [32], assuming that sensors are deployed in groups and the sensors in the same group can land in different locations following a known probability distribution. With this prior deployment knowledge, a sensor can analyze the observation of the group memberships of its neighbors, and use the Maximum Likelihood Estimation method to determine its location.

2.1.2 Secure localization schemes against wormhole attack

Y. Hu, A. Perrig, and David B. Johnson proposed the first work [50] called *pack leashes* to defend against wormhole attacks. In their work, a *temporal* packet leash is established by restricting an upper bound on the lifetime of a packet. When receiving a packet, the receiver checks if it has been expired and can discard the wormholed packet that always incurs longer processing and transmitting time. A *geographical* packet leash is established by calculating the distance between two sensors' geographical positions, thus the receiver can discard wormholed packet if it travels beyond some threshold. The limitation of temporal leash is that very precise synchronization of hundreds of nanoseconds is required, since radio signal travels at speed of light and the mutual distance between sensors are only several meters. The limitation of geographical leash is that correct geographical locations are required to establish the leash, thus it cannot be used against wormhole attacks launched to the localization process.

L. Hu and D. Evans [49] utilized sector antennas equipped on sensors to detect wormholed messages. They assume that each antenna has N equally divided zones (numbered from 1

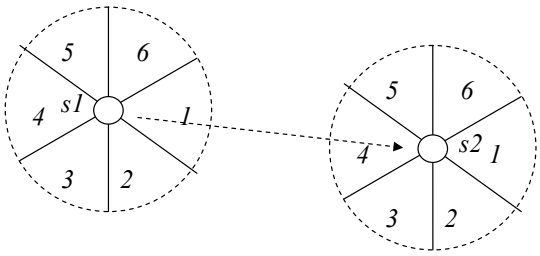


Figure 2.1 Detect Wormholes Using Sector Antennas

to N); a sensor listens to the carrier in omni mode, and receives signals through the zone on which the signal power is maximal. By using magnetic needle, it can be ensured that a particular antenna zone (e.g. zone of number 1) on all sensors always faces the same direction. From Figure 2.1, we see that the signal from true neighbors must be received in the opposite zone (zone 4) from the zone (zone 1) reported by the neighbor. However, signals received from wormholes do not always have this property. To detect more wormholes that bypass above basic checking, the authors also proposed *verified neighbor discovery* protocol and *strict neighbor discovery* protocol. These protocols all require the existence of some potential verifier nodes, thus without a verifier node, a sensor cannot distinguish legitimate neighbors from neighbors through a wormhole. This will result in the lost of some legitimate neighboring connectivity, and may lead to degradation of the localization performance.

L. Lazos and R. Poovendran proposed another secure localization scheme called *SeRLoc* [61] also using sectored antennas. Each anchor transmits different beacons at each antenna sector containing the anchor's location and the angles of the antenna boundary lines; each sensor determines its location as the center of gravity of the overlapping region of all sectors it hears. During this localization process, wormholes can be detected using two properties of SeRLoc, namely, the sector uniqueness property and the communication range violation property: if two sectors of a single anchor are heard, or if two anchors with mutual distance greater than $2R$ (R is the communication range) are heard, the sensor can detect that it is under wormhole attacks. The sensor then broadcasts a random nonce, identifies the closest anchor L_i by the first reply and takes the center of gravity closest to L_i as its estimated location, which is called

Attach to Closer Locator Algorithm (ACLA). One problem of ACLA is that innocent packets may sometimes arrive later than the ones through wormhole, because the communications are unreliable in real situations and messages may need to be re-transmitted multiple times before the receiver actually receives them.

2.1.3 Secure localization schemes against pollution attack

We observe that all attacks including fake locations, wormholes, and range modifications have a common feature, that they all provide inconsistent location references, namely, the sending sensor's location and the measured distance between the sender and the receiver are inconsistent. Therefore, some experts suggested to achieve secure localization through eliminating inconsistent references, using some statistical outlier-removing methods.

D. Liu, N. Peng, and W. Du took the mean square error (MSE) as an indicator of the degree of inconsistency among location references received by a sensor. They proposed a greedy algorithm [64] that starts with the set of all location references, and each time considers all subsets with one fewer reference and chooses the subset with the least MSE as the input to the next round, until the MSE value drops below a reasonable threshold. This scheme can effectively enhance sensors' attack-resistent ability, but also launches relatively high computation overheads on sensors. Another problem is that it requires benign references be the majority among all location references, and may not work well if corrupted location references collude together and take a larger percentage (e.g. around 50%) among all references.

Instead of identifying and eliminating inconsistent references before localization, Z. Li, W. Trappe, Y. Zhang, and B. Nath [68] proposed to live with these inconsistent references and still estimate reasonable locations for sensors using Least Median of the Squares (LMS) technique. LMS is one of the most commonly used robust fitting algorithms and can tolerate up to 50 percent outliers among the total references. Since the exact LMS solutions are computationally prohibitive, they adopt an efficient alternative technique [82] to firstly get several candidate reference subsets, then choose the one with the least median squares to estimate a sensor's location.

2.2 Location Anomaly Detection

2.2.1 Verification using special hardware

The location anomaly detection problem was firstly introduced by N. Sastry, U. Shankar, and D. Wagner [85]. The authors proposed *Echo* protocol to verify if a device is inside some specific region (e.g., a room or a football stadium) to facilitate location-based access control. Their protocol is that the verifier node sends a packet containing a nonce using RF and the device echoes the packet back using ultrasound. Then by checking the packet transmission time and the processing delay, the verifier can verifies if the device locates inside the circle region centered at the verifier. *Echo* protocol attempts to verify the presence of a device in a particular region of interest, rather than any particular *point* location.

When RF time-of-flight method is used to measure distance, distance-bounding protocol [10] can upper bound the measured distance from one device to another. The important assumption of this protocol is that the device can bound its processing (XOR) to a few nanoseconds and the verifier can measure time with nanosecond precision. Based on this distance-bounding protocol, S. Capkun and J. P. Hubaux proposed a location verification scheme for wireless sensor networks using *Verifiable Multilateration* (VM) technique [15]. The rationale behind VM technique is that when a sensor claims to locate somewhere within a triangle region formed by three verifiers, then its location can be verified if all three distances from the sensor to the verifiers are consistent with the calculated ones, because if the sensor lies about its location and meanwhile maintains the consistency, it needs to decrease at least one distance to the verifier, which cannot because of the distance bounding. The limitations of VM technique are the requirement of delicate hardware to perform distance-bounding and the requirement of dense deployment of verifiers. L. Lazos, R. Poovendran, and S. Capkun further proposed a hybrid secure localization/verification system called *ROPE* [62], combining the secure properties of *Verifiable Multilateration* technique [15] and *SeRLoc* [61].

S. Capkun, M. Cagalj, and M. Srivastava proposed another verification scheme [16] using *covert base stations*. The covert base stations (CBS) are silent to the on-going communications

and their positions are only known to the verification infrastructure. Upon receiving location messages from a sensor, several CBSs cooperate (through wired links) and check if its location is consistent with the difference of time-of-arrival to each CBS. Because sensors do not know the positions of CBS, their success rate to achieve consistency through guessing is very small. Mobile base station (MBS) can also play the role of verifier, such that it sends a verification request from one location, moves and waits for the response at a different location. Therefore, at the time of position verification a sensor does not know the position of the MBS.

2.2.2 Lightweight Verification Schemes

Unlike the above verification schemes that use some special hardware, W. Du, L. Fang, and P. Ning proposed *LAD* scheme [26] that verifies sensors' locations by checking the consistency of the locations with the deployment knowledge. They assume that all sensors are deployed in groups (each group has a unique group ID) following a known probability distribution. A sensor's location can be verified only when its neighborhood observation is consistent with that derived from the deployment knowledge. The difference of LAD from previous work is that LAD verifies all sensors that are localized within *anomaly degree* region of their true locations, rather than any precise point locations.

E. Ekici, J. McNair, and D. Al-Abri proposed *Probabilistic Location Verification* (PLV) algorithm [29] to verify sensors' locations in dense sensor networks. PLV explores the probabilistic relation between the number of hops a packet traverses to reach a destination and the Euclidean distance between source and destination. Then the verifier can determine a plausibility (between zero and one) and create a trust level for each sensor's location claim.

2.3 Secure Network Coding

The research on "secure network coding" studies how to make network coding systems secure in the presence of various malicious attacks. These attacks are generally classified into passive ones (eavesdropping or wiretapping attacks) and active ones (pollution attacks).

2.3.1 Secure Network Coding against Wiretapping Attacks

2.3.1.1 Security goals

The security goal is to prevent the source information from leaking to the adversaries without using cryptographic mechanisms such as encryptions.

We can further categorize the security goal into Shannon security and weak security. The difference between them is that Shannon security does not allow the leakage of any information about the source, while weak security does not allow the leakage of any *meaningful* information. Formally speaking, for any $M_i \in \mathcal{M}$, Shannon security requires that $H(X|M_iX = B_i) = H(X)$, while weak security requires that $H(x_i|M_iX = B_i) = H(x_i)$, for all $i \in \{1, \cdots, n\}$. As an example, given two source symbols x_1 and x_2, weak security allows the adversaries to gain $x_1 + x_2$, but Shannon security does not permit such leakage.

2.3.1.2 Shannon-secure Network Coding Schemes

Cai & Yeung [14] studied the problem of how to make a linear network coding system to transmit information "securely" in the presence of a wiretapper(or eavesdropper) who can eavesdrop on a bounded number of network links. They gave the definition of secure (i.e., Shannon-secure) network code, proposed a method to transform a given linear network code into a secure one, and presented the sufficient condition that guarantees the existence of such a secure transformation.

The basic idea of Cai & Yeung's method is to insert some random symbols into the message vector sent by the source, such that the symbols transmitted on all edges are "randomized", i.e., the message on any edge is a combination of both the information symbols and the random symbols. More specifically, the input vector at the source is divided into two portions, the first $r = n-k$ symbols are information symbols and the remaining $k = max\{k_i\}$ symbols are random symbols chosen uniformly from F_q, i.e., $X = (S, W) = ((s_1, \cdots, s_r)^T, (w_1, \cdots, w_k)^T)$. Let C denote an $n \times n$ *secure transformation matrix*. Cai & Yeung's method is to apply this matrix to an existing insecure network code and transform it into a secure one. More specifically, for any edge $e \in E$ of encoding vector $\mathcal{T}_e$, the transformed encoding vector is $\mathcal{T}'_e = \mathcal{T}_eC$. Thus,

the symbol transmitted on edge e becomes $\mathcal{T}'_e X = \mathcal{T}_e C X$. If the matrix C should satisfy two properties, namely, (1) C is full-rank and (2) the first r row vectors of matrix C^{-1} and all row vectors of matrix M_i are linearly independent, then the transformed network code is Shannon-secure, i.e., the eavesdropper cannot eliminate the randomness or learn any combinations of solely the information symbols. To guarantee the existence of such a transformation matrix C, a larger field size is required. Cai & Yeung proved that a lower bound of $q > |\mathcal{A}|$ is *sufficient*.

Feldman et al. [33] also suggested to insert random symbols into the source message, i.e., $X = (S, W) = ((s_1, \cdots, s_r)^T, (w_1, \cdots, w_k)^T)$, and tried to find a transformation matrix C to turn an existing coding scheme into a Shannon-secure one. They generalized Cai & Yeung's method and showed that finding such a matrix C is equivalent to finding a code with certain generalized distance properties, which is a Span Distance Problem. When solving the Span Distance Problem, the authors derived a *necessary* condition that can guarantee the existence of a solution to the problem. That is, the necessary lower bound of size of the finite field should satisfy $q > N^{\Omega(\sqrt{\frac{n-r}{logN}})}$, where N be the number of all edges in the network, r is the number of information symbols sent by the source and $n = r + k$ is the number of both the information symbols and the random symbols. Unlike Cai & Yeung, Feldman applied transformation matrix C to the message vector sent by the source, while keeping the code vectors on all edges unchanged. Namely, the message vector sent by the source will be $X' = CX$, and for any edge $e \in E$, the symbol transmitted on it will be $\mathcal{T}_e X' = \mathcal{T}_e C X$. This transformation is equivalent in power to that of Cai & Yeung, but it is simpler because it does not need to change the coding process on every node.

2.3.1.3 Weakly-secure Network Coding Schemes

To our knowledge, the only work to achieve weak security is done by Bhattad & Narayanan [8]. The authors observe that the security requirement [14] can be relaxed in practice and believe that it is suffice if no *meaningful* information is leaked to the computationally-bounded eavesdroppers. Unlike the Shannon-secure code, a weakly-secure code does not require any random symbols inserted into the message vector. A secure transformation matrix C can be

applied either to the message vector sent from the source or to the encoding vectors of all edges. After transformation, the message available to the eavesdropper is $M_i'X = M_iCX$ for all $M_i \in \mathcal{M}$. The authors proved that as long as matrix C satisfies two properties, namely, (1) Matrix C is full-rank and (2) any row vector of the matrix C^{-1} and all row vectors of matrix M_i are linearly independent, then the transformed network code is weakly-secure. (The reader can compare with the two properties given by Cai & Yeung, which are required under Shannon security.) The authors also gave a lower bound of the size of the finite field, which should satisfy that $q^n > |\mathcal{A}|q^k + q^{n-1}$, where k is the maximum number of edges the adversaries can wiretap, $|\mathcal{A}|$ is the total number of subsets of edges, and n is the length of the message vector. When the size q is large enough to satisfy the above inequality, the existence of the transformation matrix C can be guaranteed.

2.3.2 Secure Network Coding against Pollution Attacks

In pollution attacks, the adversaries may compromise some forwarders and modify (or pollute) their output messages. Pollution attacks not only prevent the sinks from recovering the source messages correctly, but also consume up the limited energy of the forwarders, which is especially harmful to resource-constrained wireless networks. Filtering pollution attacks is challenging in network coding systems, because traditional hashing or signature mechanisms no longer work. With traditional mechanisms, the source generates the hashes or signatures for all messages, which can be utilized by others to verify these messages. However, in network coding systems, the encoded messages are generated by the forwarders themselves and the source does not know how to produce the hashes or signatures for these encoded messages.

We classify the existing solutions for securing network coding against pollution attacks into two categories depending on whether the pollution attacks (or polluted messages) are filtered by the forwarders or only the sinks.

Basically, the solutions of the first category utilize some homomorphic function and allow the forwarders to generate and verify the hashes or signatures of encoded messages without contacting the source. Similarly, the solutions of the second category create some error correc-

tion information for the source messages. This information is either appended to the source messages or sent to the sinks in advance, and will be used by the sinks to detect or filter polluted messages. Compared with those of the second category, the solutions of the first category save energy of the forwarders by filtering polluted messages as early as possible, however, they are typically much slower due to the heavy public-key operations for generating and verifying the hashes or signatures.

2.3.2.1 Filtering Pollution Attacks at Sinks

Ho et al. [47] studied *Byzantine modification attacks* in multicast networks and illustrated how randomized network coding can be utilized to detect these attacks without the use of cryptographic functions. In Ho's scheme, the source attaches each packet with a hash calculated from a polynomial hash function. If Byzantine modification attacks exist, the sinks can detect inconsistency between the packets and corresponding hashes with high probability, as long as the sinks receive some unmodified packet whose content is unknown to adversaries. The detection probability can be traded off against communication overhead and the number of unmodified packets.

Jaggi et al. [52] discussed how to build resilient network coding in the presence of *Byzantine adversaries*. Their idea is to append the source messages with extra parity information that can be used by the sinks to correctly recover the source messages even suffering *Byzantine attacks*. The tradeoff is the sacrifice of data transmission rate. They analyzed the optimal rate that network coding can achieve under different threat models and proposed some polynomial time algorithms to attain these optimal rates. Suppose the network capacity is C. When the adversaries can eavesdrop on all links and jam z_o links, their algorithm can achieve a rate of $C - 2z_o$. In contrast, when the adversaries have limited snooping capabilities, their algorithm can achieve the higher rate of $C - z_o$.

2.3.2.2 Filtering Pollution Attacks at Forwarders

Krohn et al. [59] proposed using homomorphic hash function to verify the check blocks of a downloaded file in peer-to-peer systems, where the check blocks are linear combinations of original file blocks. Gkantsidis and Rodriguez [40] extended Krohn's approach and presented a homomorphic hashing scheme (called *GR's scheme* for short) for securing peer-to-peer file distribution systems with network coding against pollution attacks. Assuming multiple users want to download a file that is divided into n blocks $b_1, b_2, \cdots, b_n$. The source (and the system) transmits these blocks with linear network coding, that is, each forwarder transmits some encoded block $e = \sum_{i=1}^{n} c_i b_i \bmod q$, where $(c_1, c_2, \cdots, c_n)$ denotes encoding vector and q is a prime. With a homomorphic hash function, the hash of this encoded block can be represented as $h(e) = \prod_{i=1}^{n} h^{c_i}(b_i) \bmod p$, where p is another prime. Hence, if a downstream node obtains the source blocks' hashes in advance, it will be able to verify the encoded block.

Charles, Jain and Lauter [20] designed a new homomorphic signature scheme (called *CJL's scheme* for short) based on Weil pairing over elliptic curves. CJL's scheme utilizes a signature function which is linear based on some torsion points over some elliptic curve. Moreover, in CJL's scheme, the calculation of signature covers a whole augmented message. Hence, this scheme does not need any secure channel and also provides source authentication.

Zhao et al. [106] studied the content distribution applications adopting network coding and proposed a signature scheme (called *Zhao's scheme*) that allows the forwarders to filter pollution attacks. They divided a source file into multiple vectors that span a subspace. In their scheme, the source calculates a signature of the spanned subspace, then broadcasts it to all the forwarders for them to verify if a received encoded vector is in that subspace or not. To verify one vector, a forwarder should calculate $m + n$ modular exponentiations, which is the same as GR's scheme, where m is the length of each vector and n is the total number of source vectors. The authors claimed that their approach does not need extra secure channels. However, the public keys and the signature used in Zhao's scheme are both related to the downloaded file. When a lot of files need to be downloaded continuously from the source, this scheme still requires secure channels to update the public keys or signature to all forwarders.

CHAPTER 3. SECURE LOCALIZATION AGAINST ATTACKS

3.1 Introduction

To construct a secure location-aware communication system for energy-constrained wireless networks, the first and foremost goal is to help sensor nodes to estimate correct locations, which is the crucial and fundamental information for almost all applications. For example, in battlefield surveillance, a sensor that detects a tank must report the location of the event to the base station. In spatial IP address assignments, each sensor needs to know its physical location to construct its IP address. In geographical routing protocol, a sensor must know its own and its neighbors' locations in order to forward the packet to the one that is closest to the destination.

In most localization algorithms, it is assumed that a small number of *anchor*s are deployed in the field serving as location references. They can obtain their locations directly, e.g., through GPS devices. The anchors then broadcast locations in *beacon messages* to help sensors localize themselves. There are many attacks that can be launched to such localization process, including wormhole attack, sybil attack, range-modification attack, pollution attack, etc.

In wormhole attack, the adversaries copy messages at one position and replay them at another location. Hence, a sensor may receive beacon messages from a far-off anchor through a wormhole tunnel, and mistakenly think the anchor is in its neighborhood. Using the wormholed reference, the sensor will estimate a wrong location or not be able to estimate a location. Since the wormholed references are generated by legitimate anchors, the integrity of these messages are not damaged. Hence, traditional security mechanisms such as encryption and authentication are not sufficient to defend against wormhole attacks.

Defending against wormhole attacks has attracted much research interest in the past. Hu,

Perrig and Johnson described a scheme named *packet leashes* [50], in which wormholes are detected based on the constraints of a packet's transmission time or transmission distance. Hu and Evans proposed [49] to equip directional antennas on sensors, and detect wormholes when neighboring sensors are not communicating through opposite antenna sectors. Lazos and Poovendran proposed to detect wormholes [61] based on sector inconsistency or communication-range violation. This scheme requires extra communication overhead on sensors and reduced the lifetime of the network. Other researchers utilized [64, 68] statistical methods to detect and filter out inconsistent location references. These approaches, however, introduce high computation overhead to sensor nodes. Actually, since sensor nodes are limited in energy supply, memory storage and computation capacity, it is very challenging to design lightweight algorithms to defend against wormhole attacks.

Besides wormhole attack, there are many other attacks which may cause sensors to obtain a wrong location estimation. Moreover, the wrongly localized sensors will serve as reference points for other sensors and thus impact many sensors' localization. We called such attack the *Pollution Attack*. Since the proliferation of location pollution can be very fast, and many downstream sensors can be potentially affected, it is crucial to eliminate false location references as soon as possible.

In this research, we design efficient and effective techniques for securing localization and location-based services against various attackers. First, to provide secure localization against wormhole attack, we propose a Dynamic Anchor Regrouping (DAR) scheme. In DAR scheme, anchors within a local area form a group which has a unique ID and a grouping lifetime. When the grouping lifetime expires, anchors quit this group and independently join other groups. While in a group, anchors include both its group ID and its location in beacon messages and broadcast to sensors periodically. Each sensor utilizes beacon messages received from the same group to calculate its location. Second, we proposed a robust localization scheme to mitigate the impact of pollution attack. In our COnfidence-TAg (COTA) scheme, each sensor calculates a tag that quantifies the accuracy of its estimated position. Then the sensor combines its estimated location and its tag in a location reference and send to

other sensors. Upon receiving enough tagged references, an unlocalized sensor filters out bad references and computes a weighted optimal solution using the remaining ones. Both DAR and COTA schemes are affordable for low-cost energy-constrained wireless networks because no heavy communication and calculation overhead is incurred on sensors, thus the scarce energy of sensors is saved and lifetime of network can be prolonged. Furthermore, no special or expensive hardware are required by our schemes, such as precise time-measuring device or directional antennas. Simulation results show that sensors' localization accuracy and the percentages of being localized are greatly improved using our secure localization schemes.

3.2 Problem Statement

3.2.1 System Model

In our system, a small number of *anchor* nodes are deployed in the sensor field. The anchors know their locations through GPS device or manual configuration. Anchors communicate with each other with communication range l to form local groups. Meanwhile, they broadcast their locations in *beacon messages* within a transmission range R to facilitate sensors' localizations. Sensors that can directly receive beacon messages will first localize themselves. Then these sensors serve as references points and send their location references to other sensors, thus the location propagation process gets started. Current propagation techniques include DV-based method [74], [77], APS-Euclidean method [75], and Distributed Trilateration method [31], [83]. We select the most commonly used one, i.e., the distributed trilateration, as the location propagation method in our research.

3.2.2 Threat Model

When WSNs are deployed in hostile environments, the localization of sensors will be subject to many malicious attacks. We can classify the attackers into internal attackers and external attackers. Internal attacker can compromise a sensor, obtain its key materials and authenticate itself to others. External attacker cannot obtain any cryptographic secrets or authenticate

Table 3.1 Classifications of Security Attacks

	Fake Location	Wormhole	Range Enlargement	Range Reduction
Internal Attacker	•		•	•
External Attacker		•	•	•

itself, but it can corrupt the physical features of the communications between sensors. In Table 3.1, we list attack methods and the type of attackers that can perform each attack.

3.2.2.1 Fake location

Fake locations information can be generated by the internal attackers who compromise sensors and authenticate themselves as legitimate ones. The impact of this attack is twofold. First, many location-based applications such as environment monitoring and target tracking will be fooled by sensors that report fake locations, e.g., they may get wrong information about the high-temperature area or the location of enemy tank. Second, other sensors' locations will be incorrect if they refer to these fake locations to localize themselves.

3.2.2.2 Wormhole

In wormhole attack, the adversaries copy the messages heard at one location, transmit them through a tunnel and replay at another location.

Figure 3.1 illustrates how a wormhole attack can damage a sensor's localization. As shown in the figure, sensor s can directly hear the beacon message of anchor A_1, but not of anchor A_2. To attack the localization of s, an adversary establishes a wormhole between position B and C, which are near A_2 and s respectively. Then, the adversary records A_2's beacon message at position B, transmits it through the wormhole tunnel and replays it at position C. If s determines its location only based on A_2's beacon message, it may assume it is near anchor A_2, e.g., at some location s'within the transmission region of A_2. If it uses both messages of A_1 and A_2, it may either believe it is located somewhere between A_1 and A_2, e.g., at the location s'', or not able to determine its location at all because it is not expected to receive beacon messages from two anchors so far away from each other.

In such a wormhole attack, the adversaries do not need to compromise any sensor or anchor

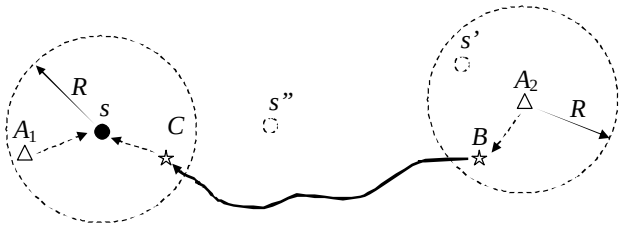


Figure 3.1 A Wormhole Attack on Sensor Localization

or to understand the meaning of the messages, they just copy and transmit the messages through the established wormhole tunnel to corrupt the localization approaches.

3.2.2.3 Range enlargement and reduction

Range modification attack is detrimental to range-based localization approaches. (1) If time-of-flight method is used to estimate distance, external attackers can intercept and replay the signal or transmit the signal through multipath, as a result, the transmitting time will be prolonged and the distance measurement will be enlarged. They can also transform the ultrasound signal into radio signal, then transform it back to ultrasound near the destination sensor, hence, the transmitting time is reduced and the distance measurement will be reduced too. For internal attackers, since they fully control the compromised sensors, they can hold the signal for a short period of time before transmitting it back to the sender, resulting in range enlargements. (2) If signal strength is used to measure distance, external attacker can strengthen or weaken the signals; internal attackers can directly broadcast signals with stronger or weaker strength.

3.2.3 Assumptions and Goals

In this research, we have the following assumptions: First, anchors cannot be compromised by adversaries, but sensors may be compromised. This assumption is reasonable in the sense that anchors are more expensive and complex hardware than sensors, and can afford advanced cryptography mechanisms to protect their secrets and privacies. Second, we assume the wormholes are bidirectional, i.e., they can transmit any message from one endpoint to the other,

and provide multiple false messages which we call *wormholed references*. However, the number of wormholed references is the minority compared with the number of benign ones in a local area. Such assumption is also adopted by previous works [64, 68].

Our research goal is to design a distributed localization scheme that can effectively defend against wormhole attack on sensors' localization. Also, when there exists any false location reference caused by wormhole attack, range modification attack, etc., we should detect the pollution and mitigate the impact to other sensors as much as possible. Our algorithms should be lightweight to be accommodated by energy-constrained networks, and they should be robust against sophisticated attacks specifically targeting our localization schemes.

3.3 Secure Localization against Wormhole Attack

3.3.1 Overview

In this section, we propose a lightweight Dynamic Anchor Regrouping scheme (DAR) against wormhole attacks on localization in wireless sensor networks. DAR consists of two schemes, namely, anchor scheme used by anchors to form groups, and sensor scheme used by sensors to determine their locations. In anchor scheme, multiple groups are dynamically formed by nearby anchors. The head of each group determines a unique ID and a random lifetime for the group. Once a group expires, each member quits the current group and independently joins other groups (or becomes a head and forms a new group if there is no group head in its neighborhood). Anchors encapsulate their locations and group IDs in beacon message to help sensors to localize. In sensor scheme, each sensor keeps receiving beacon messages and calculating location-candidates using beacon messages from the same group. It determines its final location using the candidates with the maximum weight.

Figure 3.2 illustrates the dynamic anchor regrouping using two snapshots, where A_1 to A_6 are the neighboring anchors of sensor s. In the left snapshot, A_1 to A_3 and A_4 to A_6 form two groups respectively, while in the right snapshot they form three groups at another moment. Assume sensor s uses the trilateration approach to localize itself, which means at least three beacon messages are required to calculate a potential location, then s can calculate

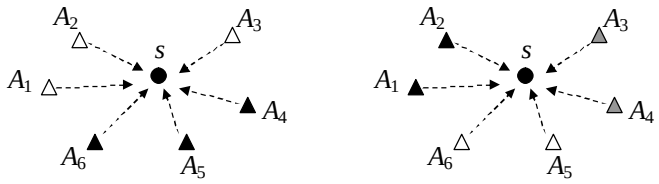


Figure 3.2 Two snapshots of dynamic anchor regrouping

two potential locations in the first snapshot, but none in the second snapshot.

3.3.2 DAR Anchor Scheme

In anchor scheme, some anchors randomly elect themselves as group heads, and broadcast regrouping messages within range l to coordinate neighboring anchors to join their groups. This range l, called regrouping range, determines the upper bound of the size of anchor groups. Note l is different from the anchor-to-sensor transmission range R which is the range to broadcast beacon messages. We first describe the details of anchors' behaviors, then analyze the determination of the regrouping range l.

After the initial *start state*, an anchor will stay in *waiting state*, *querying state* or *regrouping state* recursively during its lifetime. Figure 3.3 depicts the state transitions and the corresponding transitions rules. On the transition arrows, *"event/actions"* means that when the *event* happens, the state transition happens and the *action* is taken by anchor before entering into the next state (*"event/-"* means no actions). We explain the state transmissions in details in the following.

1. Start state:

- After deployment, each anchor stays in the initial *start state*. It immediately selects a random duration T_w from $(0, \overline{T_w})$ and enters into *waiting state*. $\overline{T_w}$ is a pre-defined maximum duration of anchors' waiting state.

2. Waiting state:

- In this state, the anchor does nothing until the timer T_w expires.

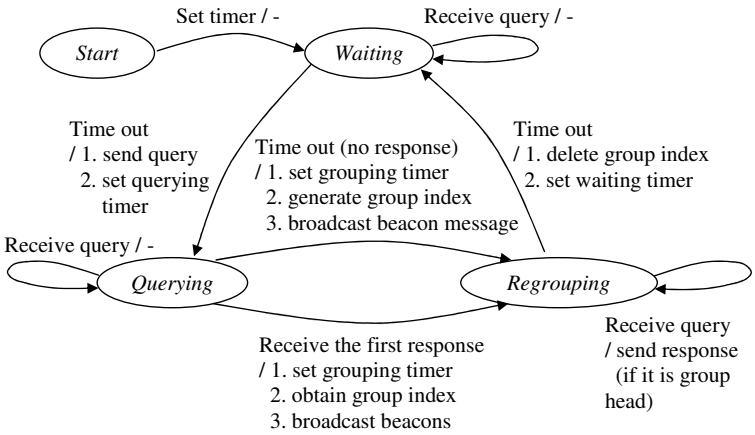


Figure 3.3 The lifetime and state transitions diagram of anchors

- When T_w expires, the anchor broadcasts querying messages within regrouping range l for any group head available in its neighborhood. Meanwhile, it sets up a timer T_q which is the duration of the querying state, then enters into *querying state*.

3. Querying state:

- In this state, the anchor waits for responds from its neighboring anchors who are group heads. The response will contain the group ID and the group's remaining lifetime.

- When receiving the first response, the anchor sets its group ID as the one in the response. Meanwhile, it sets up a timer T_r which has the value of the group remaining lifetime in the response. Then it enters into *regrouping state*.

- If no response is received and T_q expires, the anchor will elect itself as the group head and generates a group ID gId by concatenating its node index and current local time. For example, anchor i will generate a group ID as following:

$$gId = \{i,\ t_i\}, \tag{3.1}$$

where t_i is the current local time of anchor i. Then, the anchor determines the group lifetime and sets up timer T_r which is randomly selected from the range $(\underline{T_r}, \overline{T_r})$, then enters into *regrouping state*.

4. Regrouping state:

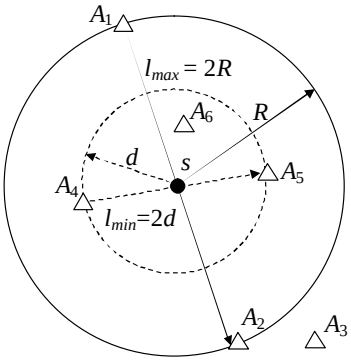


Figure 3.4 Determine l_{max} and l_{min}

- In the regrouping state, the anchor periodically broadcasts beacon message within range R to sensors.

- If an anchor is a group head and receives querying messages from other anchors, it checks if the mutual distance between the querying anchor and itself is less than l. If yes, then the head responds with its group ID and the remaining group lifetime.

- When T_r expires, the anchor deletes its group ID and returns to *waiting state* after resetting its waiting timer $T_w \in (0, \overline{T_w})$.

In our scheme, *waiting state* is introduced to prevent dismissed anchors of a group to start querying at the same time. Therefore, $\overline{T_w}$ does not need to be very long but it should allow anchors to have sufficient back-off time between their queries. Given the number of anchors and the field size, we can calculate the anchor density and estimate the average number of anchors per group, thus $\overline{T_w}$ can be properly determined. *Querying state* is used for ungrouped anchors to send request and receive response from group heads, thus the length of this state, i.e. T_q, should be longer than the round-trip time between anchors. In *regrouping state*, beacon messages are periodically broadcast to sensors. The length of this state should guarantee that all anchors broadcast beacon messages at least once before the group dismiss. Thus if the broadcasting period is a constant c, the lower bound $\underline{T_r}$ can be set as $2c$, and the upper bound $\overline{T_r}$ can be set according to the environment interferences (e.g., the beacon messages may be lost and need to be transmitted multiple times).

3.3.2.1 Group Size Control

The regrouping range l poses constraints on the size of anchor groups and affects the performance of our scheme. Only anchors within distance l have the possibility to join the same group, hence, a small l can help prevent anchors at the two endpoints of wormholes to group together. On the other side, a relatively small l would cause too few anchors to belong to a group (namely too few location references used by sensor), resulting in high localization errors of sensors. Here, we discuss how to determine the upper and lower bounds for l, i.e.,$l_{max} \geq l \geq l_{min}$.

Figure 3.4 illustrates the determination of l_{max} and l_{min}. In the figure, anchors whose beacon messages can be heard by sensor s are within the big circle of radius R. Only anchors within this range need to form the same group to facilitate s to determine its location. For instance, if A_1 is group head, it only needs to communicate with anchor as far as A_2, but not necessary to communicate with A_3 that is outside the circle. Hence, we obtain:

$$l_{max} = 2R. \tag{3.2}$$

Now we consider l_{min}. Suppose sensors use the trilateration technique and require at least three beacon messages (most other localization algorithms require less than three beacon messages), then we should guarantee the nearest three anchors have a chance to join the same group. As shown in Figure 3.4, A_4, A_5 and A_6 represent the nearest three anchors and d denotes the radius of the smallest circle containing them, we have:

$$\pi d^2 \rho = 3, \tag{3.3}$$

where ρ is the density of anchors in the field. Since the value of l_{min} should guarantee A_4, A_5 and A_6 have a chance to join the same group and the maximum distance between them is $2d$, we obtain:

$$l_{min} = 2d = 2\sqrt{3/\pi\rho}. \tag{3.4}$$

3.3.3 DAR Sensor Scheme

To determine its location, each sensor repeatedly executes two processes: (1) receiving the beacon messages; and (2) calculating the potential locations from these messages. Correspondingly, it maintains two queues for the two processes: Q_b to store received beacon messages and Q_l to store the calculated potential locations.

The length of Q_b is $\overline{T_r}$, i.e., each sensor stores the most recently received beacon messages within a period of $\overline{T_r}$. Thus, we can guarantee each sensor can maintain beacon messages from all members of any group, because $\overline{T_r}$ is the maximum lifetime of any group.

When receiving sufficient beacon messages from a group, the sensor can calculate a potential location based on these messages. The sufficient number is three if using range-based localization approaches, or one if using range-free approaches. We define the number of beacon messages used for calculation the *weight* of this potential location. The sensor stores those potential locations with the maximum weight in Q_l and uses the mean value of these locations to estimate its final location. However, if the difference between any particular potential location and the mean value is greater than twice of normal localization error of the used approach, the sensor believe it is under attacks and defers estimating the final location until some new beacon message is received.

Figure 3.5 illustrates the two queues Q_b and Q_l. In queue Q_b, the sensor records beacon messages and the corresponding times when receiving them. For example, $b_{i,2}$ is the second beacon message from group i and $t_{i,2}$ is the corresponding receiving time. In queue Q_l, only the potential locations with the maximum weight are stored. In the figure, the maximum weight is 3, and the potential location l_i is calculated using anchors from group i. The detailed procedure of sensor scheme is as follows:

1. When receiving a new message $b_{j,m}$, the sensor stores $(t_{j,m}, b_{j,m})$ in Q_b. Then it removes the messages that is $\overline{T_r}$ time units older than the newly inserted message. E.g., if $(t_{j,m} - t_{i,1}) > \overline{T_r}$, it removes $(b_{i,1}, t_{i,1})$ from Q_b.

2. For each group, the sensor counts the total number of all stored messages of group j,

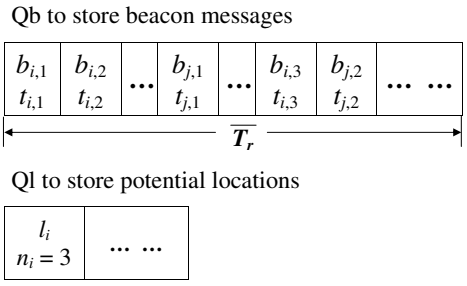


Figure 3.5 Sensor localization based on the received beacon messages

denoted by n_j:

- If n_j is less than the sufficient number of current localization approach or $n_j < n_{max}$, where n_{max} is the current maximum weight of potential locations in Q_l (initially, $n_{max} = 0$), then the sensor does nothing and goes back to wait for new beacon messages.
- If $n_j = n_{max}$, it calculates a potential location l_j based and inserts (l_j, n_j) into Q_l.
- If $n_j > n_{max}$, it clears Q_l and stores (l_j, n_j) in Q_l.

3. To estimate its final location, the sensor calculates l_{mean}, the mean value of all potential locations stored in Q_l. Then, it compares each potential location l_i in Q_l with l_{mean}. If $|l_i - l_{mean}| > 2Err$ for any particular l_i, where Err is the average localization error of the current localization approach, it stops estimating its final location and goes back to wait for new message. Otherwise, it uses l_{mean} as its final location.

In our sensor scheme, each sensor only estimates its location based on those potential locations with the maximum weight, because a larger weight implies a higher localization accuracy. When estimating a sensor's location, we assume Err is known, which is the average error of the current localization approach measured in the case of no attacks. In no-attack environment, every potential location (and also l_{mean}) should be within the range of Err from the true location, hence, the difference between any potential location and l_{mean} is at most $2Err$. If the final check fails, we assume the sensor is under attacks and its final location is undetermined.

3.3.4 Security Analysis

In our scheme, there are two kinds of messages: query and response messages communicating between anchors; and beacon messages communicating from anchors to sensors. We should ensure the authentication and integrity of these messages, such that DAR can perform properly in presence of wormhole attacks.

3.3.4.1 Wormhole Attacks against Anchor Scheme

For query and response messages, we use a common secret key that is shared by all anchors. The anchors can use this key to encrypt the query and response messages, meanwhile generate a keyed Message Authentication Code (MAC) for the messages. Since we assume the anchors cannot be compromised by adversaries, MAC is sufficient to achieve integrity and authentication.

In the anchor scheme, a group head always does the integrity check and consistency check on receiving querying messages from other anchors. That is, the head first checks the MAC to see if the querying message is generated by anchors. If yes, it further checks if the message is from an anchor who is within range l of itself. If yes, this anchor is a real neighbor; otherwise, the message must be from wormholes and will be discarded. By performing the integrity and consistency check, DAR anchor scheme guarantees that only anchors physically closed to each other can join the same group, and all wormholes larger than the regrouping range l can be filtered out. By property controlling the group size, i.e., setting the value of l within the range (l_{min}, l_{max}), most wormholes can be filtered out, especially long ones that cause severe impacts on sensors' locations.

3.3.4.2 Wormhole Attacks against Sensor Scheme

For beacon messages, we also use keyed MAC to guarantee integrity: all anchors and sensors share a common secret key k to calculate the MAC of each beacon message. To guarantee authentication, we assume each anchor (say, anchor i) can calculate a sequence of authentication keys $k_{i,1}, k_{i,2}, \cdots, k_{i,n}$ that form a hash chain from a seed authentication key

$k_{i,0}$, where n is a sufficiently large number. Namely,

$$k_{i,n} = h(k_{i,n-1}) = h^2(k_{i,n-2}) = \cdots = h^n(k_{i,0}), \tag{3.5}$$

where h is a one-way hash function. Each sensor is preloaded with the last authentication key $k_{i,n}$ for every anchor i, but the seed key $k_{i,0}$ is never exposed to any sensor. Whenever the anchor generates a new beacon message, it uses a new authentication key and encapsulates the key in that beacon message. The authentication keys are used in the reverse order as they are generated. For example, a beacon message b of anchor i is as follows:

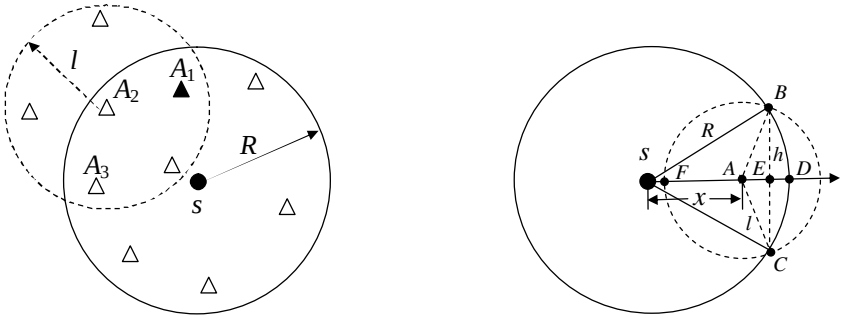
$$b = \{i,\ gId,\ (x_i, y_i),\ (j, k_{i,j}),\ MAC_k\}, \tag{3.6}$$

where gId denotes the group ID of this anchor, (x_i, y_i) is the anchor's coordination, $(j, k_{i,j})$ represent the j-th authentication key of the anchor, and MAC_k is a keyed MAC of this message calculated using key k. When a sensor receives this message, it checks the MAC to ensure the message integrity, then verifies whether $k_{i,n} = h^{n-j}(k_{i,j})$ or not. If so, it accepts the message and stores the latest key $k_{i,j}$.

Wormhole attacks can direct beacon messages from one location to another to confuse surrounding sensors. If a beacon message is from an anchor whom the sensor can directly hear, then the sensor can discard this message since the replayed message's authentication key is older than the stored one. Otherwise, if the beacon message is from a never-heard anchor, the messages are unfortunately accepted and stored in queue Q_b. However, since potential locations with the maximum weight can be stored in queue Q_l, the wormholed beacon messages in Q_b will not be used in the final location estimation, as long as the majority of location references are benign.

3.3.5 Convergence Analysis

In our scheme, a sensor cannot determine its location until sufficient number of neighboring anchors join the same group. Hence, we are concerned about the convergence property of our scheme, that is, *how long a sensor can determine its location with a given probability*? It is a really hard problem because in our scheme the anchors form groups completely in random.



(a) Calculating $p_g(m)$ (b) Calculating p_l

Figure 3.6 Calculating the probability p_l

Hence, we turn to consider a simplified scenario in which all anchors are *loosely* synchronized, i.e., they begin to construct and quit groups almost at the same time. In this scenario, a anchor's lifetime consists of many rounds that have the same durations. Thus, our question turns to calculating n_r, the number of rounds before a sensor can determine its location with a given probability p?

Theorem 1. *In the simplified scenario, a sensor needs at most* $\log_{(1-p_g)}(1-p)$ *rounds before being able to determine its location with a given probability* p*, where* p_g *denotes the probability that a sensor can determine its location in one round.*

Proof: According to the definition of p_g, it is easy to prove that

$$1-(1-p_g)^{n_r} \geq p. \tag{3.7}$$

Now, we discuss how to calculate p_g. Averagely, a sensor has $n = \pi R^2 \rho$ neighboring anchors, where ρ denotes the density of anchors in the field. Assume we randomly choose m anchors from its neighboring anchors. If these m anchors form a group and $m \geq m_0$, where m_0 denotes the sufficient number, the sensor can calculate its location. Therefore, we derive:

$$p_g = \sum_{m_0}^{n} p_g(m) = \sum_{m_0}^{n} \binom{n}{m} p_l^{\binom{m}{2}} p_h (1-p_h)^{m-1} p_f^{m-1}, \tag{3.8}$$

where $p_g(m)$ is the probability that these m anchors can form a group in one round. To form the group, these m anchors should be within the regrouping range l of each other; meanwhile,

only one can be the group head, while the other $m-1$ anchors choose the head instead of becoming head themselves. In this equation, the notations have the following meanings:

- $\binom{n}{m}$ denotes the possible selections of m anchors from the n neighboring anchors.
- p_l denotes the probability that any two of these m anchors are within the regrouping range l of each other, thus $p_l^{\binom{m}{2}}$ guarantees all m anchors can communicate with each other.
- p_h denotes the probability one anchor becomes the group head.
- p_f denotes the probability that one of the other $m-1$ anchors choose the group-head anchor (instead of becoming heads themselves).

Note: an non-group-head anchor may have some other neighbors that can also become group head. Figure 3.6(a) shows an example of choosing $m=3$ anchors from the neighboring anchors of sensor s. In the figure, A_1, A_2 and A_3 are the chosen ones and A_1 is the group head among them. The neighbors of A_2 are those within the dotted-line circle. Since these neighbors can also become group heads, A_2 has only a probability p_f choosing A_1 as its group head.

Calculate p_l: Figure 3.6(b) illustrates how to calculate p_l. In the figure, the big circle represents the region in which the neighboring anchors of sensor s may reside. Assume some anchor is at position A, whose distance from s is x. The small circle indicates the regrouping range l of this anchor. Only when another anchor resides within the overlapping area of two circles, can these two anchors join the same group. Therefore, given an anchor at position A, p_l is the ratio of the overlapping area $S_o(x)$ over the area of the big circle. Hence, we have:

$$p_l = \frac{\int_0^R \frac{S_o(x)}{\pi R^2} dx}{\int_0^R dx} = \frac{1}{\pi R^3} \int_0^R S_o(x) dx. \tag{3.9}$$

The overlapping area consists of two sectors $ABFC$ and $ABDC$:

$$S_o(x) = S_{ABFC} + S_{ABDC}. \tag{3.10}$$

It is easy to derive that:

$$S_{ABFC} = l^2(\pi - \arcsin\frac{h}{l}), \tag{3.11}$$

$$S_{ABDC} = R^2 \arcsin\frac{h}{R} - xh, \tag{3.12}$$

where

$$h = d_{BE} = \frac{[(l+x)^2 - R^2][R^2 - (l-x)^2]}{4x^2}. \tag{3.13}$$

.

From equations (3.9) to (3.12), we can calculate the value of p_l.

Calculate p_h: Each anchor competes with its neighbors within the regrouping range l to become the group head. The probability that any anchor wins the competition is:

$$p_h = \frac{1}{\pi l^2 \rho}. \tag{3.14}$$

Calculate p_f: As shown in Figure 3.6(a), a non-group-head anchor A_2 has $\pi l^2\rho$ neighbors. Among these neighbors, only one anchor (A_1) of the m selected anchors is group head. Among the remaining $\pi l^2\rho - m$ anchors, the number of heads will be $p_h(\pi l^2\rho - m)$. Therefore, A_2 has a probability:

$$p_f = \frac{1}{p_h(\pi l^2\rho - m) + 1}, \tag{3.15}$$

to choose A_1 as its group head.

In summary, p_g can be calculated from equations (3.8), (3.9), (3.14) and (3.15). Thus, we conclude that in this simplified scenario a sensor can be localized with a given probability p within at most $n_r = \log_{(1-p_g)}(1-p)$ rounds. In the following section, we study the normal scenario by simulation.

3.4 Secure Localization against Pollution Attack

3.4.1 Overview

In this section, we propose a secure localization scheme, named COTA, to defend against pollution attack. As shown in Figure 3.7, during the localization process, each sensor stays in one of the states including *waiting state*, *localizing state*, and *transmitting state*. A sensor initially stays in waiting state and receives reference messages from its neighbors. As soon as

the adequate number (which is three if using trilateration technique) of references are received, it enters the localizing state and performs COTA scheme. COTA consists of a localization phase and a tag-generation phase. In the first phase, a sensor filters out bad references according to the absolute and relative metrics, then checks if the remaining references are more than the minimum number. If there is not sufficient references, it goes back to waiting state for more references; otherwise, it calculates its position through weighted optimization mechanism. In the second phase, the sensor firstly computes the statistical or geographical indictors, then derives its localization error and translates it into a confidence tag. Finally the sensor enters transmitting state, combines its estimated position and the confidence tag into a reference message and sends it to others.

In our scheme, a concept called *Confidence Tag* is used. It indicates the reliability of a sensor's estimated position, and is calculated using the tag-generation function.

Definition 1. *Let p_e and p_t be a sensor's estimated position and true position, thus $e = |p_e - p_t|$ is its localization error. We call function $t = f_t(e)$ the* ***tag-generation function****, and call t the* ***confidence tag*** *of the sensor. $t \leq T$ is a nonnegative integer, where T is the highest confidence tag, e.g., $T = 8$.*

Definition 2. *Let e and t be the localization error and the confidence tag of a sensor, and $\hat{e}$ is an upper bound of e, we call function $\hat{e} = f_e(t)$ the* ***inverse-tag function****.*

We note that function f_e is not the exact inverse function of f_t, because instead of returning a sensor's localization error e from its tag t, f_e returns an upper bound $\hat{e}$. The coefficients of these functions can be computed and stored in sensors memories, and they are used through COTA scheme.

3.4.2 COTA Scheme: Localization Phase

The localization phase of COTA consists of two function blocks: reference filtering and weighted optimization. We use the tuple (p_i, t_i, d_{ij}) to denote the location reference sent by node i and received by node j, where p_i and t_i are the claimed position and the confidence tag of node i, and d_{ij} is the mutual distance measured by sensor j.

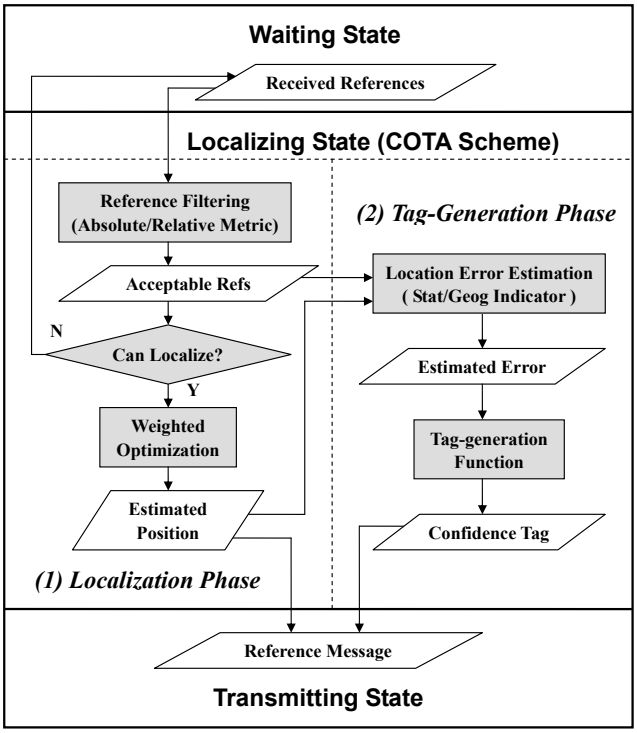


Figure 3.7 Sensor States & COTA Scheme

3.4.2.1 Filtering Metrics

We provide two filtering metrics: the absolute metric and the relative metric. When using the absolute metric, sensor simply filters out bad reference whose confidence tag $t < t_0$. The threshold t_0 can be obtained through training. Let e_{max} be the maximum localization error of sensors in non-adversary scenarios, then $t_0 = f_t(e_{max})$ is the minimum reasonable confidence tag. If t_0 is set very high, then most references will be filtered out and a sensor may not acquire adequate number of references to localize. Therefore, we need to consider the tradeoff between sensors' localization error and the localized percentage. Actually, how to set a reasonable threshold is application-specific.

The relative metric is computed by $u = f_e(t)/d$. A sensor filters out bad reference if $u > u_0$, where u_0 is a preset threshold. We use a figure to illustrate the underlying idea. In Figure 3.8, sensor s has a reference message (p_a, t_a, d_{as}). We can compute an upper bound of node a's localization error by function $\hat{e_a} = f_e(t_a)$. Namely, a's claimed position p_a should be within

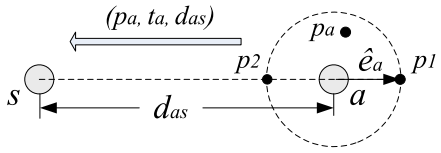


Figure 3.8 COTA Filtering Metric: Relative Metric

distance $\hat{e_a}$ of its true position. Hence, the distance from position p_a to sensor s should be no larger than $d_{as}+\hat{e_a}$ and no smaller than $d_{as}-\hat{e_a}$. The relative metric $u = f_e(t)/d = \hat{e}/d = \Delta d/d$ essentially indicates the relative accuracy of the mutual distance, and the references with higher uncertainties than the threshold will be dropped. We can obtain u_0 through training. If sensors' maximum localization error in non-adversary scenarios is e_{max} and the communication range between sensors is R, then we set the threshold by $u_0 = e_{max}/R$.

The two filtering metrics can be used separately or together. In our simulation, it shows that the absolute metric outperforms the relative one in providing stronger filtering capacity, but realizes less localized percentage. However, when both metrics are applied, sensors can be localized with better performance than that when using a single metric.

3.4.2.2 Weighted Optimization

After filtering out the bad references, each sensor uses the remaining references and performs the trilateration technique to compute its position. Because of the noisy rang measurements and various attacks to the location references, the unique solution may not exist to satisfy all the constraints. In COTA, we use Weighted Least Square Estimation (WLSE) mechanism to compute the optimal solution for each sensor.

Assume sensor s has n references (p_i, t_i, d_{is}), where $p_i = (x_i, y_i)$, $0 \leq i \leq n$. Sensor s weighs the references by their confidence tags and computes the optimal solution (x_0, y_0) by minimizing the following equation:

$$Min \sum_{i=1}^{n} t_i^2 \cdot (\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_{is})^2 \tag{3.16}$$

This nonlinear LSE optimization problem can be solved by many standard methods, e.g., the MMSE matrix solution [37] or Kalman filter method [12], [84]. In COTA, we adopt the

MMSE technique and transmit the nonlinear problem into linear equations:

$$\begin{aligned} t_1 \cdot \sqrt{((x_1 - x_0)^2 + (y_1 - y_0)^2)} &= t_1 \cdot d_{1s} \\ t_2 \cdot \sqrt{((x_2 - x_0)^2 + (y_2 - y_0)^2)} &= t_2 \cdot d_{2s} \\ \cdots &\quad \cdots \\ t_n \cdot \sqrt{((x_n - x_0)^2 + (y_n - y_0)^2)} &= t_n \cdot d_{ns} \end{aligned} \tag{3.17}$$

After squaring and rearranging terms on each side of equations 3.17, we obtain:

$$\begin{aligned} 2t_1^2 x_1 x_0 + 2t_1^2 y_1 y_0 &= t_1^2(x_1^2 + y_1^2 - d_{1s}^2 + x_0^2 + y_0^2) \\ 2t_2^2 x_2 x_0 + 2t_2^2 y_2 y_0 &= t_2^2(x_2^2 + y_2^2 - d_{2s}^2 + x_0^2 + y_0^2) \\ \cdots &\quad \cdots \\ 2t_n^2 x_n x_0 + 2t_n^2 y_n y_0 &= t_n^2(x_n^2 + y_n^2 - d_{ns}^2 + x_0^2 + y_0^2) \end{aligned} \tag{3.18}$$

Then we compute the average of above equations:

$$C_x \cdot x_0 + C_y \cdot y_0 = C_d + x_0^2 + y_0^2, \tag{3.19}$$

$$\begin{aligned} C_x &= 2\sum_{i=1}^{n} t_i^2 \cdot x_i^2 / \sum_{i=1}^{n} t_i^2, \\ C_y &= 2\sum_{i=1}^{n} t_i^2 \cdot y_i^2 / \sum_{i=1}^{n} t_i^2, \\ C_d &= \sum_{i=1}^{n} t_i^2 \cdot (x_i^2 + y_i^2 - d_{is}^2) / \sum_{i=1}^{n} t_i^2 \end{aligned}$$

We multiply equation (3.19) by t_i^2 and subtract it from equations in (3.18). Finally, we get following standard linear least square equation:

$$A \cdot [x_0 \ y_0]^T = B, \tag{3.20}$$

$$A = \begin{bmatrix} 2t_1^2 \cdot (x_1 - C_x/2) & 2t_1^2 \cdot (y_1 - C_y/2) \\ 2t_2^2 \cdot (x_2 - C_x/2) & 2t_2^2 \cdot (y_2 - C_y/2) \\ \vdots & \vdots \\ 2t_n^2 \cdot (x_n - C_x/2) & 2t_n^2 \cdot (y_n - C_y/2) \end{bmatrix}, \quad B = \begin{bmatrix} t_1^2(x_1^2 + y_1^2 - d_{1s}^2 - C_d) \\ t_2^2(x_2^2 + y_2^2 - d_{1s}^2 - C_d) \\ \vdots \\ t_n^2(x_n^2 + y_n^2 - d_{ns}^2 - C_d) \end{bmatrix}$$

Then the optimal matrix solution can be given by:

$$[x_0 \ y_0]^T = (A^T A)^{-1} A^T B \tag{3.21}$$

3.4.3 COTA Scheme: Tag-generation Phase

In the tag-generation phase, a sensor estimates its localization error then generates a confidence tag. Since sensor has no knowledge of its true position, it needs some indicators to derive its localization error. In this subsection, we firstly propose two indicators: the statistical indicator and geographical indicator, then discuss the construction of the tag-generation function and inverse-tag function.

3.4.3.1 Statistical Indicator

We use the weighted sum of error squares, called residual r, as the statistical indicator of sensor's localization error:

$$r = \sum_{i=1}^{n} t_i^2 \cdot (\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} - d_{is})^2 / \sum_{i=1}^{n} t_i^2 \tag{3.22}$$

Consider that all the references are correct and accurate, namely, both the references' positions and mutual distances are correct, then the residual r will be minimized to zero and the sensor will be localized at its true position. It suggests that small residuals may imply consistent references and accurate location estimations for sensors. Thus, we intend to obtain an increasing function f_s which can map the residual r to sensor's localization error $\tilde{e}$ by $\tilde{e} = f_s(r)$.

We do experiments in non-adversary scenarios to explore the relationship between residual r and sensor's localization error e. However, we notice that small residuals sometimes lead to large localization errors. Such results are caused by the noisy distance measurements. In presence of measuring errors, incorrect solution may sometimes better minimize the residual than the true one. This *flex ambiguity* problem in localization is studied by David Moore et al. in [69], in which the authors also proposed a geographical constraint called *Robust quadrilateral* to the location propagation process. Localized sensors and one unlocalized sensor form several triangles. If all triangles satisfy $b \cdot sin^2\theta > d_{min}$ (where b is the length of the shortest side, θ is the smallest angle, d_{min} is a preset threshold), this quadrilateral is considered to be robust and can be used for location propagation. We add this constraint to the simple trilateration

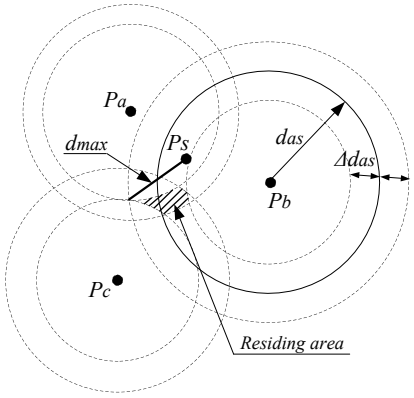


Figure 3.9 COTA Localization Error Indicator: Geographical Indicator

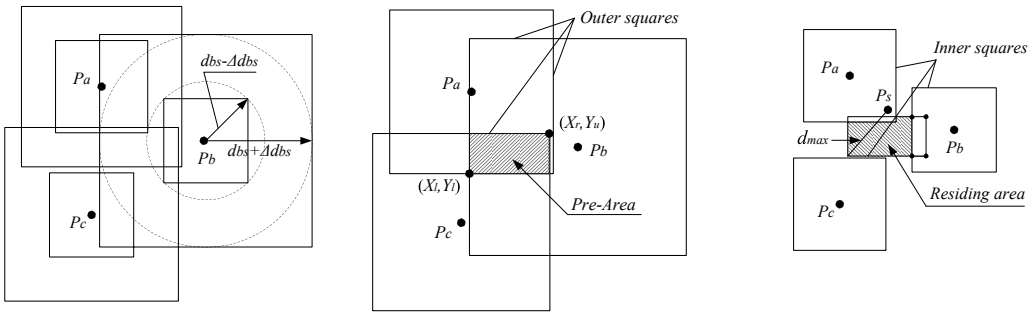
technique and find that the relationship turns more regular. In our simulation setup, we calculate the function coefficients as $\tilde{e} = f_s(r) = 2.5 * r + 1.5$.

3.4.3.2 Geographical Indicator

The statistical indicator requires the function coefficients to be computed offline and stored in sensors' memories. In this subsection, we propose a geographical indicator d_{max} that can be computed on-the-fly after sensors are deployed in the field.

As shown in Figure 3.9, sensor s has three references: (p_a, t_a, d_{as}), (p_b, t_b, d_{bs}), (p_c, t_c, d_{cs}). The location error of each reference node can be estimated by $\hat{e_i} = f_e(t_i)$, and the location uncertainty $\hat{e_a}$, $\hat{e_b}$, $\hat{e_c}$ can be translated into the uncertainty of the mutual distance, given that the claimed positions p_a, p_b, p_c are correct. Also consider the distance measurement error δd_{is}, we can obtain the overall distance error by $\Delta d_{is} = \hat{e_i} + \delta d_{is}$. Therefore, sensor s should reside inside several rings, each of which centers p_i and whose inside/ouside radiuses are $d_{is} - \Delta d_{is}/d_{is} + \Delta d_{is}$. We call the overlapping region of all these rings as *residing area* (the shadow area). Since s's true position is within this area, thus the maximum distance d_{max} from its estimated position p_s to this area is an upper bound of its localization error, which we take as the geographical indicator.

In what follows, we discuss how to obtain the geographical indicator d_{max} efficiently. Since the computation is expensive to determine *residing area* based on the intersection of rings, we



(a) Step 1 (b) Step 2 (c) Step 3

Figure 3.10 COTA Geographical Localization Error Indicator: Computation Steps.

replace each ring with a "frame" and simplify the computation process in following three steps (Figure 3.10).

Step 1: Replace each ring with a frame, whose outer/inner squares are tangent with the outer/inner circles of the ring.

Step 2: Obtain the overlapping region of all the outer squares, called *pre-area*. Its lower-left and upper-right coordinates are (X_l, Y_l) and (X_r, Y_u):

$$(X_l, Y_l) = (\max_{i=1\cdots n}\{x_i - \Delta d_{is}\}, \max_{i=1\cdots n}\{y_i - \Delta d_{is}\})$$
$$(X_r, Y_u) = (\min_{i=1\cdots n}\{x_i + \Delta d_{is}\}, \min_{i=1\cdots n}\{y_i + \Delta d_{is}\})$$

However, because of the noisy distance measurements and attacks, we may have $X_l > X_r$ or $Y_l > Y_u$ in above equations. Thus, we make following relaxations to the coordinates:

$$If\ X_l > X_r,\ X_l' = \min_{i=1\cdots n}\{x_i - \Delta d_{is}\}, X_r' = \max_{i=1\cdots n}\{x_i + \Delta d_{is}\}$$
$$If\ Y_l > Y_u,\ Y_l' = \min_{i=1\cdots n}\{y_i - \Delta d_{is}\}, Y_u' = \max_{i=1\cdots n}\{y_i + \Delta d_{is}\}$$

Step 3: Exclude from the *pre-area* any parts inside the inner squares, resulting in the *residing area*. Since the maximum distance from a point to a convex polygon is between the point and one of the polygon's points, we only record the points' coordinates. When the *pre-area* intersects with an inner square, their overlapping region should be excluded (Figure 3.10(c)), thus some points will be deleted and some new points will be introduced. On comparing the distances from p_s to each of the points, we can obtain the maximum distance d_{max}.

3.4.3.3 Tag-generation function

Each sensor uses the tag-generation function $f_t(e)$ to translate its estimated localization error into a proper confidence tag. $f_t(e)$ should have the following properties:

- Decreasing function: if $x_1 > x_2$, then $f_t(x_1) < f_t(x_2)$. This guarantees high localization errors will generate low confidence tags.
- The domain of $f_t(e)$ are sensors' localization errors, namely an infinite rational range $(0, +\infty)$.
- The range of $f_t(e)$ are sensors' confidence tags, namely a set of discrete nonnegative integers $\{0, 1, \cdots, T\}$.

Since decreasing function $f_t(e)$ maps an infinite interval to a finite set of nonnegative integers, thus when $e > e_0$, the function should output the lowest confidence tag zero. We can construct a linear function as following:

$$t = f_t(e) = \max(0, \lfloor -\frac{T}{e_0} \cdot e \rfloor + T), \tag{3.23}$$

where e_0 is the boundary value that $f_t(e_0) = 0$, T is the highest confidence tag.

We perform simulations in non-adversary scenarios to obtain a reasonable boundary value e_0. Since more than 90% sensors can be localized with $e < 10m$, we set $e_0 = 10m$ and believe that a sensor with localization error larger than that has a high probability to be attacked. We set $T = 8$ in our experiments. The value of T is application-specific, and higher T generally leads to more delicate differentiation between localization accuracies. The inverse-tag function $f_e(t)$ is closely related to $f_t(e)$. We provide the following construction:

$$\hat{e} = f_e(t) = -\frac{e_0}{T} \cdot (t - T), \tag{3.24}$$

where $\hat{e}$ is an upper bound of localization error e.

3.4.4 Security Analysis

Besides launching conventional attacks, the adversaries may corrupt COTA scheme by performing some specific attacks, e.g., manipulating sensors' confidence tags. Therefore, we

must consider the existence of corrupted reference with disguised tag, i.e., the tag does not correctly represent the reliability of the location information. At following, we list four types of attacks that an adversary may perform, from simple ones to more sophisticated ones:

- Decrease-tag Attack: A compromised sensor can broadcast its location reference with a smaller tag than the true one. Since it's worth nothing to decrease the confidence tag of a false reference, which may cause it to be filtered out or bear small weight during localization, the adversary probably launch this attack to the correct location references.

- Remain-tag Attack: A compromised sensor or wormhole transmitter can produce incorrect location references whose confidence tags are kept unchanged.

- Increase-tag Attack: This attack is the opposite to the decrease-tag attack, in which the adversary broadcasts location references with higher confidence tags.

- Invert-tag Attack: Sophisticated adversaries can invert tags: set low tags to correctly localized sensors and high tags to wrongly localized sensors.

For the decrease-tag attack, the resulting small-tag references will probably be filtered out before participating in localization, thus a sensor may fail to localize itself for lack of enough valid references. The adversary performs this attack to launch the denial of service (DOS). However, since they can simply jam the communication between sensors or destroy sensor nodes to launch DOS, delicately tampering the confidence tags cannot achieve any extra benefit.

In the remain-tag and increase-tag attacks, a sensor may estimate a false position, because incorrect references are not filtered out and may be high-tag attached. However, if the sensor being attacked can generate a low confidence tag to its location reference, it will be dropped by its neighbors and will not affect many other nodes. Therefore, COTA is robust in the sense that it prevents local damage from proliferating to other areas. The reason why wrongly localized sensors can generate low confidence tags relies in the effective indicators: as long as there are some benign references inconsistent with the incorrect ones, the statistical indicator (residual) will produce big values, and the geographical indicator will result in large residing area and a long distance d_{max}. Then the tag-generation function (decreasing function) can compute

Table 3.2 Simulation Parameters and Default Values

	Meaning	Default
FM	the filtering metric	Combination
EI	the localization error indicator	Statistical
D	the damage degree	20m
n	the number of false references	1
P	the percentage of sensors being attacked	10%
n_f	the noise factor of distance measurement	1%

low tags from the large indicators. The increase-tag attack is more severe than the remain-tag attack, thus we simulate it in our simulations. We further assume that the adversaries always generate the highest confidence tag to the false references. Experiment results show that COTA can survive through various attacks and provide accurate location estimations.

The invert-tag attack is the most sophisticated, in which the adversary not only contaminate correct references by attaching low tags, but also disguise false references with high tags. The impact is that the victim sensor not only wrongly localizes itself but also computes a high confidence tag to its reference. However, to mount this attack, an attacker should be very resourceful to jam/tamper all the benign references and launch multiple false references. Furthermore, the attacker needs to launch invert-tag attack to each victim sensor, otherwise the victim sensor will performs as increase-tag attacker who can affect only the direct neighboring nodes.

3.5 Simulation Study

3.5.1 DAR Scheme Localization Performance

3.5.1.1 Simulation Setup

We deploy 4,800 anchors and 10,000 sensors uniformly and randomly in a $1000m \times 1000m$ square area. The transmission range of beacon messages is set as $R = 20m$. Each sensor can hear 6 anchors on average. We create wormholes by choosing their source and destination endpoints randomly in the sensor field. These wormholes replay beacon messages with the same broadcast range R as real anchors to sensors. The upper bound of anchors' waiting period is set by $T_w = 10s$; the lower and upper regrouping period are set by $\underline{T_r} = 30s$ and

$\overline{T_r} = 100s$. We simulate the noisy distance measurement by adding Gaussian noise to the real distance: $\tilde{d} = d \cdot (1 + x * f)$, where $\tilde{d}$ and d are the measured and the real distance, x is a standard normal random number, f is the noise factor which is set to 5% in our simulation.

We compare sensors' localization performances using and not using DAR scheme on top of several localization approaches. The approaches we test include both range-free ones and range-based ones. In Centroid approach, a sensor estimates its location as the mean value of its neighboring anchors' locations. In Overlapping Circles (OC) approach, a sensor estimates its location as the Center of Gravity (CoG) of the overlapping region of the transmission ranges of its neighboring anchors. In Overlapping Sectors (OS) approach, which is proposed in [61], anchors broadcast beacon messages in sectors using their equipped directional antennas, and a sensor estimates its location as the CoG of the overlapping region of the sectors of its neighboring anchors. Notice in OS approach, we adopt SeRLoc localization method, but use DAR scheme to defend against wormholes. In Trilateration (Tri) approach, a sensor uses least mean square technique to determine its location.

3.5.1.2 Metrics and Parameters

The metrics we use to evaluate the localization performances of sensors include the localization error and the localization percentage, which are defined in the following:

$$LE_o = \frac{1}{N} \sum_{i=1}^{N} \sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2} \tag{3.25}$$

$$LP_o = N_l / N \tag{3.26}$$

where (x_i, y_i) and $(\tilde{x}_i, \tilde{y}_i)$ are the real and the estimated locations of sensor i. N is the total number of sensors deployed in the field, and N_l is the number of sensors that are capable to localize themselves (the locations of the remaining $N - N_l$ sensors are undetermined).

We also study the localization error and the localization percentage of the sensors being attacked, to further study the direct impacts of wormhole attacks on the victims. The metrics

are defined by:

$$LE_k = \frac{1}{N_k}\sum_{i=1}^{N_k}\sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2} \quad (3.27)$$

$$LP_k = N_{kl}/N_k \quad (3.28)$$

where N_k is the number of sensors being affected by the wormholes, and N_{kl} is the number of victims who are able to localize themselves.

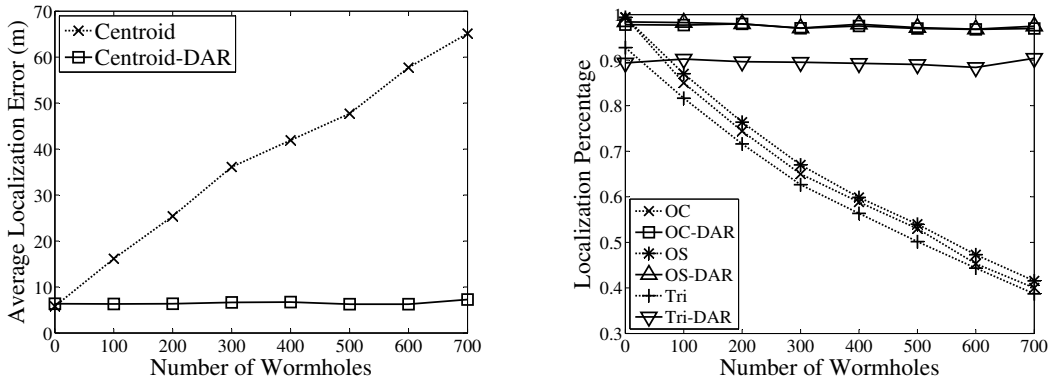
To study DAR scheme's resistance to wormhole attacks in different situations, we vary the following parameters:

- n: the number of wormholes launched in the field.
- l: the regrouping range of anchors.
- T_l: the total running time of DAR scheme.
- λ: the length of wormholes.

In our simulation, we deploy each wormhole by choosing their two endpoints randomly in the sensor field, and vary the parameters n, l and T_l each at one time to test its impact on the localization performance of sensors. The default values are $n = 500$, $l = 40m$ and $T_l = 400s$. Then, we vary the length λ to study the impact of wormholes with different lengths.

3.5.1.3 Impact of wormhole numbers n

In this subsection, we increase the number of wormholes from 0 to 700, and compare the localization performances of sensors before and after using DAR scheme. Figure 3.11(a) shows the average localization error LE_o when using Centroid approach. In the case of no protection, LE_o increases from $7m$ to $70m$; using DAR scheme, LE_o always remains around $7m$. Figure 3.11(b) shows the localization percentage LP_o when using Centroid. In Centroid, a sensor always estimated its location as the average of its neighboring anchors' locations (even though they are under attacks), thus wormhole attacks do not impact the localization percentage at all. When using DAR, the percentage is slightly decreased, but it is still above 97%. The



(a) Localization Error

(b) Localized Percentage

Figure 3.11 Impact of number of wormholes when using Centroid method

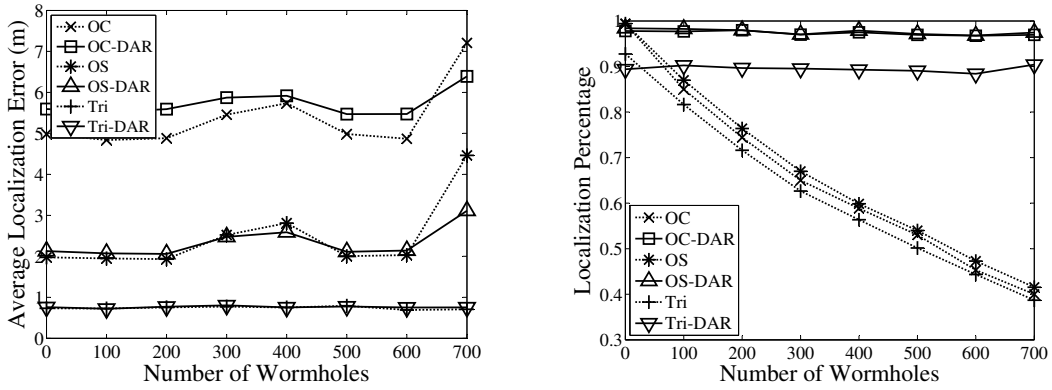
reason is that some sensors detect they are under attack and remain unlocalized rather than accepting a corrupted location estimation.

Figure 3.12 shows the localization performance when the sensors localize themselves using Overlapping Circles (OC), Overlapping Sectors (OS), and Trilateration (Tri) approaches. In Figure 3.12, we see that localization errors are not severely impacted by wormhole attacks, while the localization percentage is greatly reduced, e.g., in most approaches, it drops from around 90% to around 40%. The reason is because sensor being attacked cannot obtain an overlapping region or compute a Least-Mean-Square solution for its location. When DAR is used, the percentage is maintained at the same level as that in the non-attack situations. Therefore, based on Figure 3.11 and Figure 3.12, we can conclude that DAR can effectively mitigate the impacts of wormhole attacks on sensors' localizations.

3.5.1.4 Impact of regrouping range l

In this subsection, we deployed 500 wormholes in the field. Using equation(3.4) in Section III-C, we can compute that the minimum regrouping range is $l_{min} = 14m$. We increase l from $14m$ and study sensors' localization errors and localization percentage.

In Figure 3.13(a), we observe that when l is around $35m$, sensors' localization errors in different localization approaches all reach their corresponding optimal value and keep the



(a) Localization Error

(b) Localized Percentage

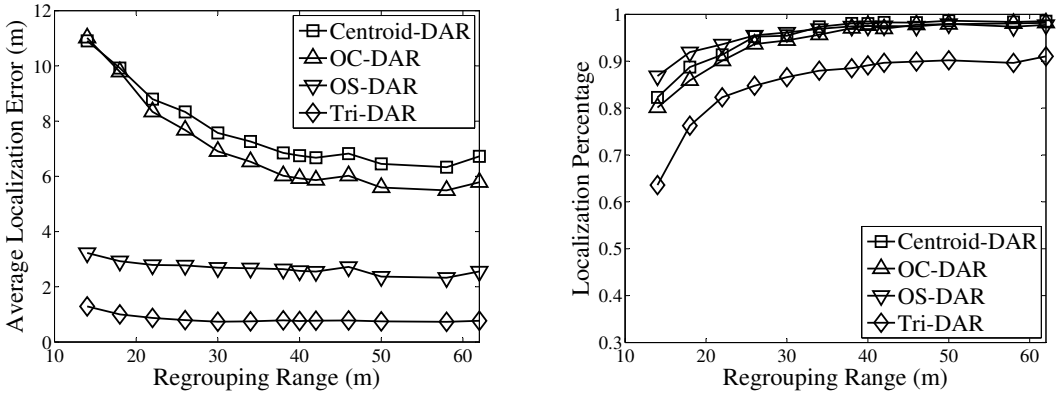
Figure 3.12 Impact of number of wormholes when using Overlapping Circles (OC), Overlapping Sectors (OS), Trilateration (Tri) methods

value afterwards. The reason that small regrouping range incurs high localization errors are twofold: First, anchors are more physically closer to each other when l is small. This will lead to large overlapping regions in range-free approaches and unbalanced distribution of location references in range-based approaches, both resulting in low localization accuracy. Second, the number of anchors of the same group decrease when l is small, which leads to less location references and less location accuracy. In Figure 3.13(b), we observe similar tendencies. That is, when l is larger than $35m$, sensors' localization percentage can be optimized.

This experiment shows us that DAR scheme is not sensible to the value of regrouping range l: When l varies within a large range $[30m, 60m]$, we can achieve the optimal localization performance for sensors using DAR scheme.

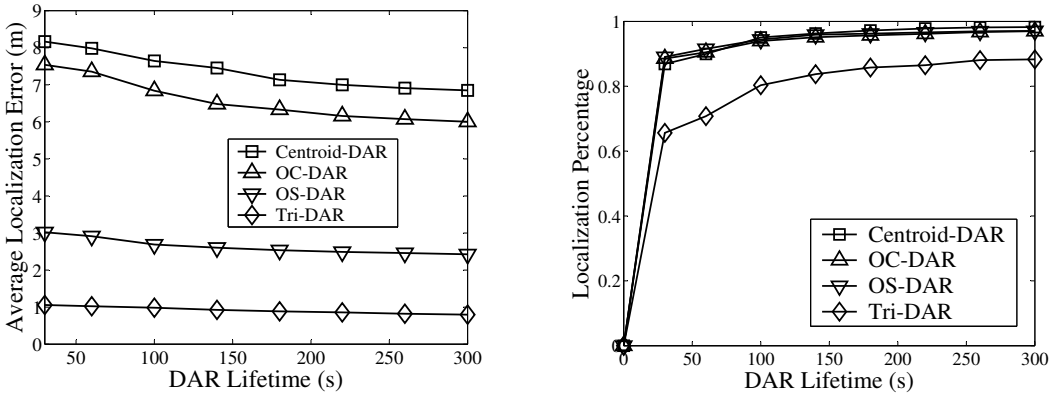
3.5.1.5 Impact of running time T_l

In DAR scheme, anchors keep on regrouping and broadcasting beacon messages, and sensors keep on localizing themselves whenever they receive sufficient number of beacon messages. If the localization phase lasts longer, on one side, more sensors should be localized, resulting in higher localization percentage; on the other side, sensors can calculate more potential locations for themselves, thus localization errors should be improved too. We study the convergence



(a) Localization Error (b) Localized Percentage

Figure 3.13 Impact of regrouping range



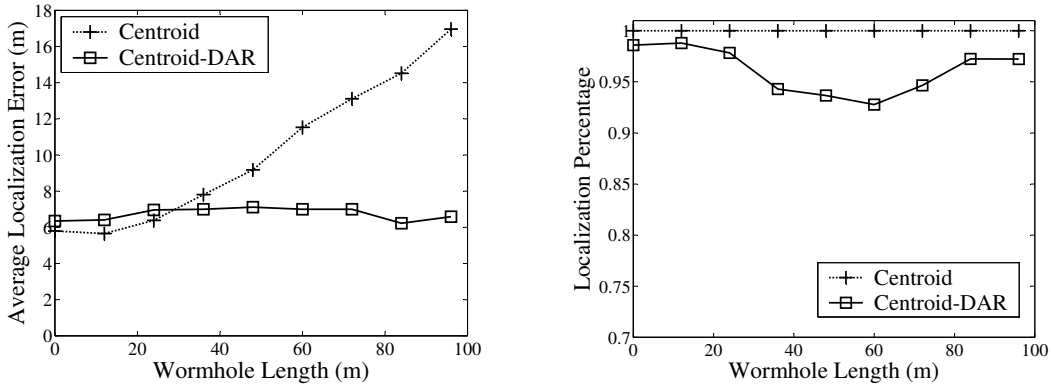
(a) Localization Error (b) Localized Percentage

Figure 3.14 Impact of running time

property of DAR scheme in simplified scenario in Section V, now we study the convergence property in normal scenarios. Figure 3.14 shows the experiment results. We see that both localization error and percentage reach their respective optimal values around $T_l > 150s$. Notice the average group lifetime is $(\underline{T_r} + \overline{T_r})/2 = 65s$, which means DAR can converge only after about two regrouping rounds.

3.5.1.6 Impact of wormhole length λ

To study the impact of different-length wormholes to the victim sensors, we vary wormhole length λ from $0m$ to $100m$ and record the value of localization error LE_k and localization



(a) Localization Error

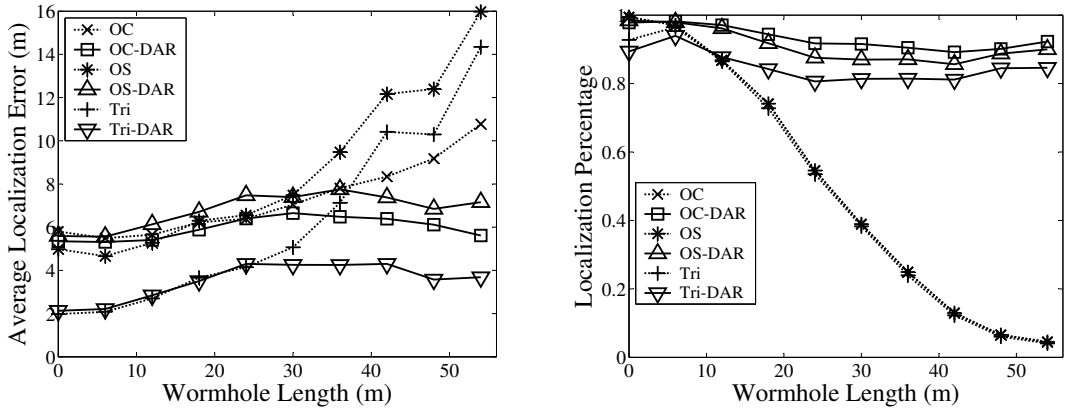
(b) Localized Percentage

Figure 3.15 Impact of length of wormholes when using Centroid method

percentage LP_k.

Figure 3.15 shows the results when sensors use Centroid as the localization approach. We observe that when no protection is applied, localization error sharply increases; when using DAR scheme, the localization error is bounded by less than $1m$ compared to that in non-attack scenarios. As we see in Figure 3.15(b), wormhole attacks do not impact the localization percentage when using Centroid, because a sensor always compute a location for itself no matter the location is corrupted or not. When DAR scheme is used, localization percentage is lower-bounded by 92%. This is because wormholes whose lengths are not very long are difficult to be filtered out, thus a sensor will detect its being attacked and choose to make its location undetermined, rather than localizing itself with large errors.

Figure 3.16 shows us the localization performance when sensors use OC, OS and Tri localizaton approaches. We see in Figure 3.16(b) that when using OC, DAR scheme can achieve an upper bound of $6m$ for the localization error; when using OS and Tri, the upper bound is only $4m$. Such localization errors are close to those ($4m$ for OC, $2m$ for OS and Tri) in non-attack scenarios. In Figure 3.16(b), we see that when no protection exists, wormhole attacks cause great impacts on the localization percentage: When wormholes are longer than $50m$, fewer than 0.5% sensors being attacked are capable to localize themselves. In contrast, DAR scheme helps maintain the localization percentage LP_k over 80% for all localization approaches.



(a) Localization Error

(b) Localized Percentage

Figure 3.16 Impact of length of wormholes when using Overlapping Circles (OC), Overlapping Sectors (OS), Trilateration (Tri) methods

3.5.2 COTA Scheme Localization Performance

3.5.2.1 Simulation Setup

In our simulation, 600 sensors are deployed uniformly and randomly in a square field of $300m \times 300m$. The sensor-to-sensor and anchor-to-sensor communication range are both $R = 25m$, thus each sensor can hear 12 neighbor nodes on average. Within the center square area, whose lower-left/upper-right coordinates are $(100, 100)/(200, 200)$, we randomly deploy 10 anchors, thus about 18 (3%) sensors can receive adequate number of beacon messages to be firstly localized. Anchors' positions bear the highest confidence tag $T = 8$. We perform the non-protected localization scheme and COTA in every deployment, and average the results over 100 independent deployments. The distance measurement error model is $\widetilde{d} = d + d * x * n_f$, where $\widetilde{d}$ and d are measured and real distance, x is uniformly distributed within $[-1, 1]$, n_f is the noise factor. E.g., if $n_f = 1\%$, then $|\widetilde{d} - d| \leq 1\% \cdot d$.

Various attacks (cheating report/wormholes/rang enlargement/range reduction) will result in either *false position* or *false range* attack. To launch the former attack, we change one of sensor s's location references from (p_i, t_i, d_{is}) to (p'_i, T, d_{is}). To launch *false distance*, we replace the reference with (p_i, T, d'_{is}), where d'_{is} is smaller than the communication range R, otherwise it will be easily detected. We perform the tag-increase attack to COTA by appending the

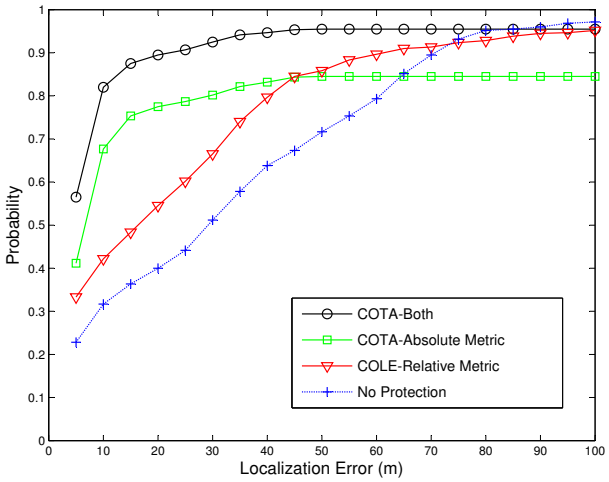


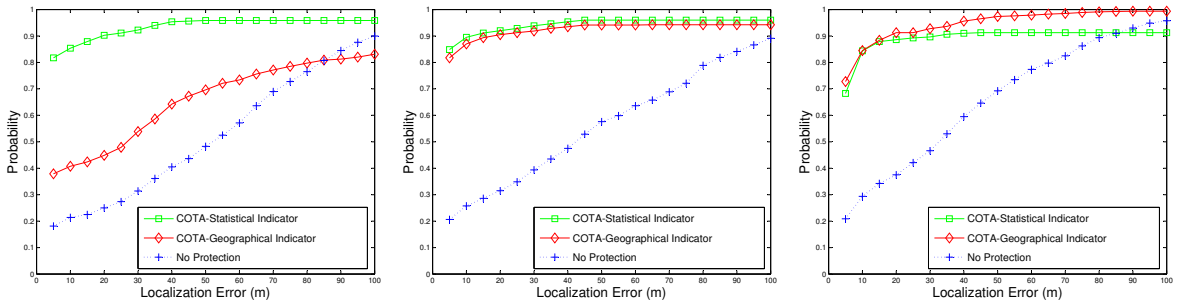
Figure 3.17 CDF of Localization Error using different FM

highest tag T to each false reference. As we have discussed, this attack is both powerful and easy to perform. We also perform non-colluding and colluding attacks by launching different numbers of false references to target sensors. In the former scenario, only one of the references that sensor s uses is contaminated; in the latter scenario, multiple contaminated references will collude to mislead s to localize at another position.

3.5.2.2 Filtering Metric (FM)

The goal of this experiment is to study the localization performance of COTA with different filtering metrics (FM). Figure 3.17 shows the CDF of sensors' localization errors when using the unprotected multi-hop scheme and COTA. We can see that without protection, more than 50% sensors are localized with $L_e > 30m$, which indicates that local damage has proliferated and impacted many sensors. When using COTA, the localization performance is effectively improved, e.g., more than 80% sensors are localized with $L_e < 10m$ when the two filtering metrics are applied together.

The two filtering metrics have different performances. Firstly, absolute metric can localize more sensors with small errors, e.g., about 70% sensors can be localized with $L_e < 10m$ when using the absolute metric, but only 40% when using the other. The reason is that absolute metric makes a sensor to discard all inaccurate references whose location errors are higher



(a) $n = 1$ (b) $n = 3$ (c) $n = 5$

Figure 3.18 *CDF* of Localization Error using Different Localization Error Indicators (EI)

than e_{max}, but the relative metric fails to do so if the false reference contains a large mutual distance, which leading to a small $u = f_e(t)/d < u_0$. Secondly, the absolute metric has a low localized percentage: when the CDF curve turns flat, only 84% sensors are localized. By looking at sensor field snapshot, we found it is because many peripheral sensors are unlocalized for lack of enough valid references.

When the two metrics are used together, COTA has the best performance and achieves both small L_e and high L_p. Therefore, we use the two metrics together to filter out bad references in the rest of our studies.

3.5.2.3 Localization Error Indicator (EI)

The goal of this experiment is to study how the localization error indicator (EI) affects the performance of COTA.

Figure 3.18 shows the CDF of sensors' localization errors when the number of false references are $n = 1, 3, 5$. In Figure 3.18(a), the statistical indicator outperforms the geographical one: more sensors can be localized with the same localization error; when $n = 3$, they have very similar performances; when $n = 5$, the geographical indicator performs a little better. For example, in Figure 3.18(c) about 92% sensors can be localized with $L_e < 20m$ when using the geographical indicator, higher than the 88% when using the statistical one. The reason is that when there are overwhelming consistent false references, the sensor will localize itself at

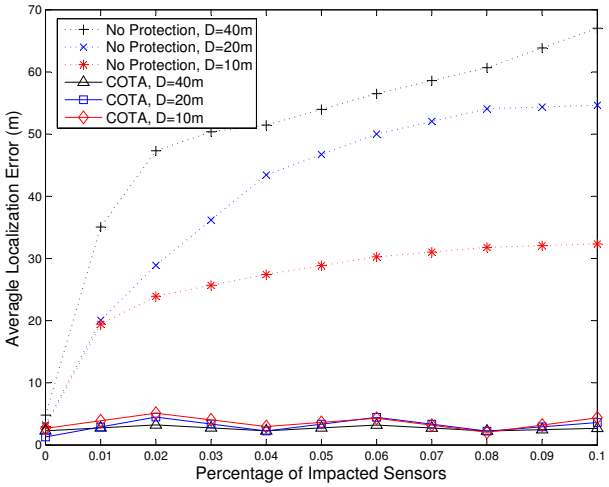


Figure 3.19 $\overline{L_e}$ under Different Attack Percentages and Damage Degrees

a wrong position with a small residual, thus the statistical indicator can not properly indicate sensor's localization error. But the geographical indicator always produces large values of the residing area and d_{max}, as long as benign references can be received. Therefore, we conclude that the statistical indicator is more effective in defending against non-colluding attacks, but the geographical indicator performs slightly better against colluding attacks. We adopt the statistical indicator as default in our simulations.

3.5.2.4 Attack Percentage (P) and Damage Degree (D)

In this subsection, we test the robustness of COTA under different attack percentages (P) and damage degrees (D). We observe in Figure 3.19 that when damage degree is set as $D = 20m$ and the attack percentage P increases from 1% to 10%, the average localization error $\overline{L_e}$ of non-protected localization scheme increases quickly from $20m$ to $50m$. Secondly, $\overline{L_e}$ also increases with the damage degrees. E.g., when the attack percentage is set as $P = 2\%$, and $D = 10m$, $20m$, $40m$, the localization error $\overline{L_e}$ is $22m$, $28m$, $48m$ respectively. COTA effectively improves the localization performance: $\overline{L_e}$ is less than $5m$ in all the cases.

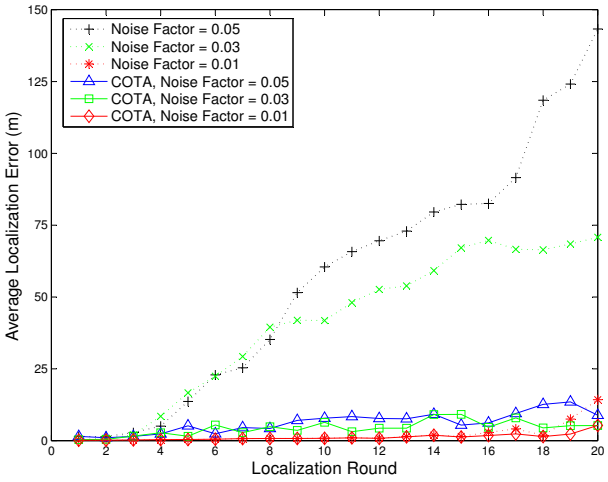


Figure 3.20 $\overline{L_e}$ under Different Noise Factors

3.5.2.5 Noise Factor n_f

The goal of this experiment is to test COTA's property of mitigating error accumulation problem caused by noisy distance measurements in non-adversary scenarios. We vary the noise factor n_f and study the average localization error.

The simulation results are shown in Figure 3.20. We compute $\overline{L_e}$ of sensors that are localized in each round. "Round" roughly indicates the localizing sequence of sensors. In the first round, the sensors who can receive adequate number of beacon messages from anchors are localized. Then the second-round sensors will be localized using location references from the first-round sensors and the anchors, and so forth. We can see from the figure that when noise factor is 1%, error accumulation problem is not severe that sensors in all rounds can be localized with small errors. However, when $n_f = 3\%$, L_e grows quickly as the round increases; when $n_f = 5\%$, the problem becomes more severe. COTA can greatly mitigate the error accumulation problem, that even in the last round, the average localization error is less than $10m$, which means the sensors who reside around the peripheral areas in the sensor field can still be accurately localized.

3.6 Conclusion

Location information is very important in communication messages in wireless sensor networks. To obtain correct locations estimations in presence of malicious attacks, we proposed several secure location schemes in this research. The first scheme DAR is a lightweight dynamic anchor regrouping scheme. It relies on local coordination of anchors to defend against and filter out wormhole attack. In DAR, anchors independently communicate with their neighbors to form groups, and encapsulate their locations and group IDs in beacon messages. Each sensor calculates location candidates after receiving sufficient number of beacon messages from the same group. Each candidate has a weight which is the number of references used to calculate this location. Finally, the sensor estimates its final location based on candidates with the maximum weight. DAR is lightweight by moving the computation overhead from sensors to anchors. In addition, it requires no special hardware such as precise time-measuring device or directional antenna. By carefully controlling the group size, DAR efficiently and effectively defends against wormhole attacks, i.e., it provides low localization errors and high localization percentage for sensor nodes. Second, we proposed a secure localization scheme COTA to defend against pollution attack. COTA is based on the novel notion of confidence tag which quantifies the accuracy of sensor's location estimation. We evaluated the localization performance of COTA through simulations. It shows that COTA can effectively prevent local location errors from proliferating to other sensors.

CHAPTER 4. DETECTING ANOMALIES IN LOCATION CLAIMS

4.1 Introduction

When the sensor networks are deployed in hostile environment such as battlefield, sensor nodes are subjected to various attacks. For example, they may be captured and compromised by adversaries, and report false information to the base station. To build a location-aware communication system, it is import to detect location anomalies which can cause severe consequence. For example, when military sensor networks are used to monitor enemy movements and suspicious phenomenons, a detection report with wrong location of a tank/bomb can cause significant damage.

There has been lots of research efforts on designing location anomaly detection techniques. We classify these research into two categories, namely, *on-spot* verification and *in-region* verification. On-spot verification is to verify whether sensors' true locations are within a certain error range from their true locations. Locations that cannot be verified are considered as anomalies. To obtain the desired on-spot verification results, these algorithms either utilize the deployment knowledge of sensors in the field [26] or make use of some dedicated hardware to verify distance measurements [11, 15, 16, 62]. For example, in [16], it is assumed that some *covert base stations* are deployed through out the field. These special base stations communicate with each other through wired links and purposely hide their existences from being discovered by sensors. Then these base stations can verify sensors' locations by checking whether the distances calculated using sensors' estimated locations are the same as the distances they directly measure using RF signals. In [11, 15, 62], it is required that sensors are able to measure time in nanoseconds in order to detect range reductions that directly impact localization results. Since existing verification algorithms either require deployment knowledge or expensive hard-

ware that may not be unavailable in low-cost wireless sensor systems, a lightweight verification algorithm is needed to efficiently and effectively perform on-spot verifications.

Besides the on-spot verification, research efforts were also devoted to designing in-region location verification algorithms. Shankar and Wagner first defined the in-region verification concept in [85]. They proposed a protocol named *Echo* to verify if a sensor is inside a physical region such as a room, a building, a sport stadium, etc. Based on the verification result, access right can be properly assigned to sensors (i.e., people who take the sensors) to access some resources in that physical region. As the first work, *Echo* successfully utilizes in-region verification to facilitate location-based access, however, it can not be directly applied in other location-based applications, because the verification region may not be explicit and we need to determine carefully by analyzing applications' functions. Secondly, when performing in-region verification, *Echo* requires the use of multiple *verifiers* that can transmit radio signal and receive ultrasound signal, and bound their XOR operations within the magnitude of nanoseconds. Such verifiers increase the expense and requires extra deployment efforts, and they are not always available in all wireless networks.

In this research, we designed several anomaly detection algorithms that overcomes the shortcomings of previous research. We combine effectively correlated sensor data to reveal location anomalies. First, two algorithms are proposed to provide on-spot verification, namely, the Greedy Filtering by Matrix (GFM) algorithm and the Greedy Filtering by Trustability-indicator (GFT) algorithm. Second, a probabilistic algorithm is proposed to perform in-region verification. All algorithms in this research are lightweight because they do not require any dedicated hardware or infrastructures, and they do not incur high computation overhead at the sensor side, so that our verification algorithms can be applied to energy-constrained wireless networks. Moreover, they are robust in presence of malicious attacks, which are launched by sophisticated attackers and try to exploit the system's weakness.

4.2 Problem Statement

In this research, we intends to design verification algorithms which efficiently and effectively determine if sensors' claimed locations are trustable. More precisely, on-spot verification verifies whether a sensor's estimated location is away from its true location less than a certain distance; In-region verification, on the other hand, verifies whether a sensor is within a geographical region given that its estimated location is in that region. If a location passes the verification, then it is recognized as a correct location; otherwise, it is an abnormal location.

The verification algorithms should have following properties. First, they should be lightweight in terms of hardware cost and computation overhead. They should not require expensive equipment such as directional antennas, radio signal transmitters/receivers, or fast processors that performs XOR computation within several nanoseconds. They should not incur high communication overhead on sensor side, which would quickly consume the scares energy stored at sensors. Second, the algorithms should be effective by achieving high detection rate and low false alarm rate. The former rate is defined as the ratio between the numbers of detected wrong locations and the number of all wrong locations, while the second rate is defined as the ratio between the number of correct locations that are mistakenly recognized as wrong ones, and the number of all correct locations. Third, as the verification algorithms may become the target of attackers, they should be robust and capable to provide decent verification results even in presence of malicious attacks.

In the following, we fist describe our system model and attack model, then we describe on-spot verification and in-region verification problems in more detail.

4.2.1 System Model

In our system, all sensor nodes can estimate their locations in the field using any of the existing localization schemes. These locations are called sensors' *estimated* or *claimed* locations, and the distances between sensors' estimated locations and true locations are called the *localization errors*. The *communication range* of a sensor is a circle centered at the sensor's true location with a certain radius. We assume all sensors' communication ranges have the

same radius. Each sensor broadcasts its ID within its communication range, and passively overhears IDs broadcast by other sensors. We say sensor A can *observe* sensor B, if A can receive the ID message from B. And we call the list of IDs that a sensor observes the sensor's *neighborhood observation*. Notice that environmental interruptions and permutations exist, so that neighborhood observation is not always symmetric. For example, sensor B may not be able to observe sensor A when A can observe B.

Our system is consisted of ordinary sensors and a Verification Center (VC) that verifies if sensors' estimated locations are trustable. The VC resides at the base station or control center, and can be safely protected from attackers. Each sensor reports its estimated location and its neighborhood observation to the VC. We assume each sensor shares a pairwise key with the VC, so they can encrypt the message and authenticate themselves. Such pairwise keys can either be preloaded off-line into sensors' memories, or distributed online using some existing key distribution algorithms [18, 25, 27, 66]. Finally, any routing protocol may be potentially used to route sensors' reports to the VC except the location-based routings, because sensors' locations are not trustworthy and wrong locations will lead to loops or even delivery failures.

4.2.2 Attack Model

We consider both passive and active attack models. Passive attackers can eavesdrop on the communications of sensors, or create wormholes [50] between two sensors that are far apart, so the sensors will mistakenly believe that they are neighbors. Active attackers can compromise sensors and send false information to its neighbors. Because active attackers can obtain all the secret information of the compromised nodes, they can deceive the VC as they were benign ones. Moreover, according to Kerckhoff's principle, we assume that the attackers know the verification mechanism, thus they can purposely launch attacks that exploit the weakness of the system. To deceive the VC into accepting wrong locations, multiple attacks can even collude together. The only assumption we make about the attackers is that in a local area, the attackers are not the majority compared with benign ones, which is also an assumption held by pervious works. For example, if a sensor has five neighbors and as long as less than three of

them are compromised ones, then the VC could still correctly verify the sensor's location. We notice such attacks are very expensive to launch because the attacks need to compromise many sensors in order to distort one location estimation. We leave the study of defending against local-dominating attackers as our future research.

4.2.3 On-spot Location Verification

On-spot verification is to verify whether a sensor's localization error is less than a certain distance. Let L_{true} and L_{est} denote the true location and the estimated location of a sensor. The verification passes if the following condition holds true: $|L_{true} - L_{est}| \leq D$, where D is named the *Anomaly Degree*. The value of D should be set properly with the considerations of the application requirements and the value of "normal" localization errors in no-attack environment. In this paper, we consider D as an input parameter and assume its value has already been specified.

4.2.4 In-region Location Verification

We use some examples to explain in-region verification. The first example is the location-based access control applications [85], in which the privilege to access some resources is granted to people who is physically inside a geographical region, e.g. a room. People are attached with sensors that communicate with the VC to get themselves verified and granted the access right.

Another example is the battlefield surveillance application. In such applications, sensors are deployed in a battlefield to monitor enemy's behaviors and report suspicious phenomenon such as appearance of tanks or soldiers. A field snapshot is shown in Figure 4.1. The symbols S and S' denote a sensor's true and estimated location. At some time, the sensor detects a tank and notifies the control center. Before the control center projects a bomb to destroy the tank, it fist consults the VC whether the estimated location of the sensor is trustable, so that it can have some confidence whether the tank can be wrecked or not. Given the bombing range, the VC first derives the verification region, which is depicted with lattice pattern in the figure. If the VC verifies that the sensor is in this region, then it implies that the tank

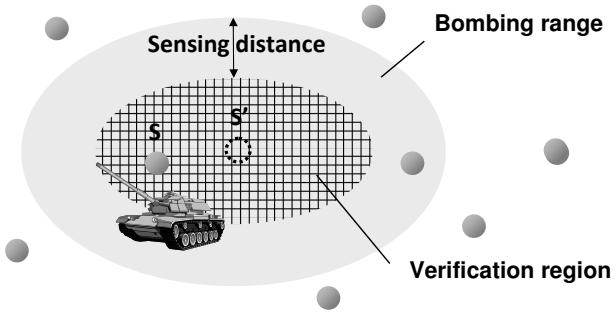


Figure 4.1 Battlefield surveillance application

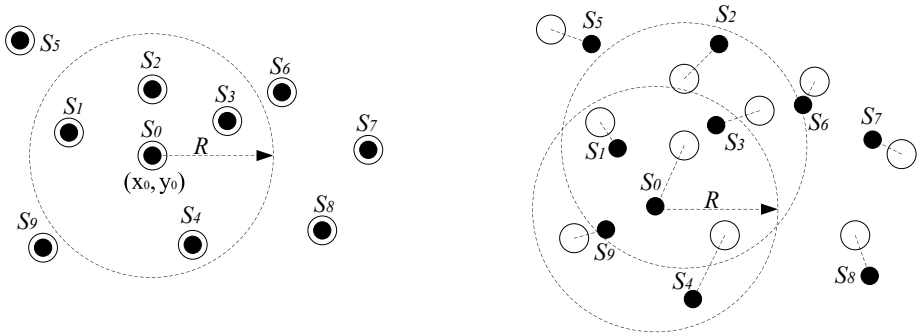
is in the bombing range, because the sensor and the tank are within the sensing distance so that the sensor detected the tank. In this example, the verification region is not explicitly given. In Section 5.1, we will discuss the determination of the verification region for different applications.

4.3 On-spot Location Verification

In this section, we propose two algorithms that the VC can use to perform on-spot location verification. The first one is named *Greedy Filtering using Matrix (GFM)*; the second on is named *Greedy Filtering using Trustability-indicator (GFT)*. Both algorithms utilize the inconsistency between sensors' estimated locations and neighborhood observations. They can be used in different scenarios according to the application's requirements.

4.3.1 Greedy Filtering using Matrix (GFM)

The first step in the verification process is that each sensor broadcasts its ID within its communication range and meanwhile overhears the IDs broadcast by other sensors. We denote sensor S_i's the neighborhood observation by O_i. As an example, Figure 4.2(a) shows a scenario where sensors are localized accurately with zero errors. The solid circles and the hollow circles represent sensors' true and estimated locations respectively. Sensor S_0's true location is $L = (x_0, y_0)$ and its communication range is the big dashed circle. Because sensor S_1, S_2, S_3, S_4 are in the communication range of sensor S_0, their ID messages can reach S_0. Hence, sensor S_0's



(a) With no localization errors (b) With localization errors

Figure 4.2 Snapshot of sensor field

neighborhood observation is $O_0 = (S_1, S_2, S_3, S_4)$. Then, each sensor sends its neighborhood observation and its estimated location to the VC. The VC will analyze all the information collected from sensors and detect if there is any inconsistency.

The intuition of GFM algorithm is that when sensors are correctly localized with small localization errors, then their neighborhood observations should be *consistent* with their estimated locations. For example, in Figure 4.2(a), all sensors are localized with no error. The distance between the estimated locations of S_0 and S_1 is less than the radius R, which is consistent with the fact that they can observe each other. Based on this intuition, GFM algorithm organizes all the information in the form of matrix to find inconsistencies among them.

4.3.1.1 Constructions of Matrixes

Suppose there are totally n sensor nodes in the filed denoted by $S_1, \cdots, S_n$. For convenience, we assume sensor S_i's ID is integer i where $i \in \{1, \cdots, n\}$. In GFM algorithm, five $n \times n$ square matrixes are calculated based on the reported information from sensors.

- **Observation Matrix:** This matrix is computed using sensors' neighborhood observations. Elements in this matrix are either 1 or 0 depending on whether sensors can observe each other, namely:

$$M_o(i,j) = \begin{cases} 1, & \text{if sensor } S_i \text{ observes } S_j \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where $i, j \in \{1, \cdots, n\}$ are the row and column index. Note that because of environmental disruptions, two sensors may not observe each other at the same time, so matrix M_o is not a symmetric matrix.

- **Estimation Matrix:** This matrix is computed using sensors' estimated locations. If the distance between S_i's and S_j's estimated locations is less than R, the radius of communication range, then the element at row i and column j will be 1, otherwise, it will be 0, namely:

$$M_e(i,j) = \begin{cases} 1, & \text{if } d_{ij} \leq R \\ 0, & \text{if } d_{ij} > R \end{cases} \tag{4.2}$$

where d_{ij} denotes the distance between the estimated locations of sensors S_i and S_j.

- **Difference Matrix:** This matrix is calculated by XORing the observation matrix and the estimation matrix:

$$M_d = M_o \bigoplus M_e, \tag{4.3}$$

where $\bigoplus$ means element-wise XOR operation. Nonzero elements in this matrix are caused by sensors' localization errors. For example, when $M_o(i,j) = 1$ and $M_e(i,j) = 0$, then $M_d(i,j) = 1$. In this example, sensor S_i observes sensor S_j, which implies the mutual distance between their *true* locations is in the communication range, however, because of localization errors, the mutual distance between their *estimated* locations is larger than the radius of communication range. Figure 4.2(b) further demonstrates how such inconsistencies are generated. In the figure, sensors' true locations are depicted as solid circles, and their estimated locations are depicted as hollow circles. We can observe that S_0's and S_2's estimated locations are within communication range, but their true locations are not, so they cannot observe each other. We will utilize matrix M_d to detect large localization errors. But prior to that, we should study that when sensors are localized with small errors (errors smaller than the anomaly degree D), how "inconsistent" the matrix M_d looks like.

- **Weight Matrix:** In our experiment, we randomly deploy 1200 sensors in the field. Sensors' communication range is $R = 20m$ and the anomaly degree is given by $D = 10m$. All sensors are localized with errors that uniformly distribute in the range of $[0, 10m]$. We calculate that given the value of d_{ij} (the distance between S_i's and S_j's estimated locations), the probability that their true locations are within communication range. The results are shown in Figure 4.3(a). We observe that when d_{ij} grows from $0m$ to $40m$, the probability drops from 1 to 0 gradually. We fit the scatter points using a simple segment-wise function $f(x)$. Furthermore, we derive another function $F(x)$ named "weight function" using $f(x)$:

$$F(x) = \begin{cases} f(x), & \text{if } x \leq R \\ 1 - f(x), & \text{if } x > R \end{cases} \tag{4.4}$$

  As shown in Figure 4.3(b), the value of $F(x)$ indicates the probability that sensors' reported information is consistent. More precisely, when $x = d_{ij}$ is greater (smaller) than $R = 20m$, $F(x)$ is the probability that the mutual distance between their true locations is greater (smaller) than R. We can also consider the value of $F(x)$ as the "weight" of the consistencies, and obtain the weight matrix M_w using the weight function, namely:

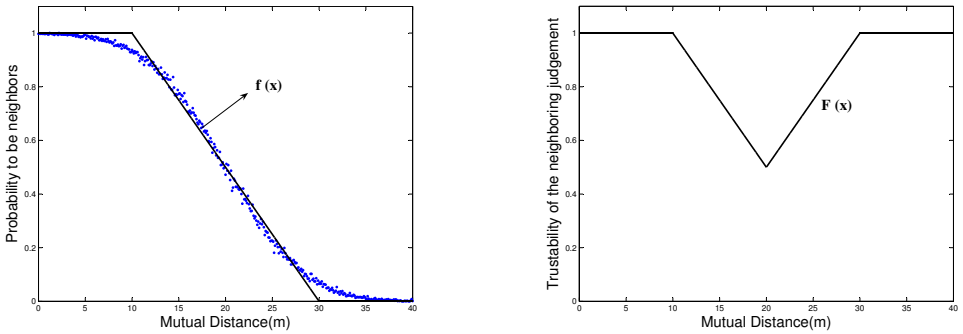
$$M_w(i, j) = F(d_{ij}), \tag{4.5}$$

  where $i, j \in \{1, 2, \cdots, n\}$, and d_{ij} is the distance between S_i's and S_j's estimated locations.

- **Inconsistency Matrix:** We multiply each element in the difference matrix with the corresponding element in the weight matrix, and obtain the inconsistency matrix M_{inc}:

$$M_{inc} = M_d \bigodot M_w, \tag{4.6}$$

  where $\bigodot$ denotes element-wise multiplication. We will utilize this matrix to *greedily* filter out the locations that cause greater inconsistencies than other locations. Notice that for a sensor S_i where $i \in \{1, 2, \cdots, n\}$, the inconsistencies brought by this sensor are represented by the nonzero elements in the i^{th} row and the i^{th} column in matrix M_{inc}. In the following, we first define several metrics that are used in the filtering process.



(a) Statistical results (b) Weight function F(x)

Figure 4.3 Weight function

4.3.1.2 Metric for filtering Abnormal locations

- Active Difference Metric:

$$AD_i = \sum_{k=1}^{n} M_{inc}(i,k), \tag{4.7}$$

where $i \in \{1, 2, \cdots, n\}$. For a sensor S_i, metric AD_i is the sum of elements in i^{th} row of matrix M_{inc}. This metric quantifies the inconsistency between sensor S_i's neighborhood observation and the estimated locations.

- Passive Difference Metric:

$$PD_i = \sum_{k=1}^{n} M_{inc}(k,i), \tag{4.8}$$

where $i \in \{1, 2, \cdots, n\}$. For a sensor S_i, metric PD_i is the sum of elements in i^{th} column of matrix M_{inc}. This metric quantifies the inconsistency between other sensors' observation on S_i (namely, sensor S_i is *passively* observed) and the estimated locations of sensors.

- Asymmetry Metric:

$$AS_i = \sum_{k=1}^{n} |M_{inc}(i,k) - M_{inc}(k,i)|, \tag{4.9}$$

where $i \in \{1, 2, \cdots, n\}$. In non-attack environment, sensors' observations are not symmetric due to environmental disturbance. However, if such asymmetry is greater than

a threshold (an extreme case is that a sensor can observe all its neighbors, but none of these neighbors can observe itself), then there may be some anomalies.

- Consistent-Neighbor Metric:

$$CN_i = \sum_{k=1,k\neq i}^{n} M_o(k,i) \times M_e(k,i), \tag{4.10}$$

where $i \in \{1, 2, \cdots, n\}$. This metric counts the number of a sensor's consistent neighbors. Here we define that a sensor S_k is a consistent neighbor of sensor S_i if it can observe S_i and its estimated location is in the communication range of S_i's estimated location.

4.3.1.3 Greedy Filtering Procedure

In this section, we describe how GFM algorithm calculates all the above matrixes and utilizes filtering metrics to greedily filter out abnormal locations.

The procedure is shown in Figure 4.4. In the first round, VC computes matrix M_{inc} and metrics AD_i, PD_i and AS_i for all $i \in \{1, 2, \cdots, n\}$. If there is any sensor whose metric value exceed that metric's threshold, VC revokes the sensor that has the largest metric value (say node S_k), and sets all zeros to the k^{th} row and the k^{th} column in matrixes M_e, M_o and M_{inc}. This process repeats until no more sensors can be filtered out. Then the metric CN_i is considered: sensors that do not have enough number of consistent neighbors are revoked. Finally, the remaining sensors are accepted by the VC as correctly-localized sensors.

In the above procedure, the threshold for different metric can be obtained through off-line training using experimental data. In our simulations, we deploy sensors randomly in a square field and localize them with errors less than the anomaly degree. Then we compute all the matrixes and the values of AD_i, PD_i, AS_i and CN_i for all sensors. The threshold value is determined according to the desired false alarm rate. For example, if the application requires that the false alarm rate should be smaller than 5%, then we set the thresholds for metric AD, PD and AS at the 95% percentile and the threshold for metric CN at the 5% percentile. Given the value of communication range radius, the node density, the accuracy of the localization

Compute matrix M_{inc} (1)

Compute metrics AD, PD, AS for all sensors (2)

While (sensor S_i exists that can be filtered out) (3)

  if $AD_i > AD$-threshold (4)

    revoke S_k that AD_k is the largest among all sensors (5)

  else if $PD_i > PD$-threshold (6)

    revoke S_k that PD_k is the largest among all sensors (7)

  else if $AS_i > AS$-threshold (8)

    revoke S_k that AS_k is the largest among all sensors (9)

  set zeros to k^{th} row and k^{th} column in M_e, M_o, M_{inc} (10)

  recompute metrics AD, PD, AS for all sensors (12)

While (sensor S_i exists that $CN_i < CN$-threshold) (13)

  revoke sensor S_i for not having enough neighbors (14)

Remaining sensors are accepted (15)

Figure 4.4 The Greedy Filtering by Matrix (GFM) Algorithm

algorithms used in the filed and the environmental parameters, we can construct experiments accordingly and obtain the above threshold values.

4.3.1.4 Security Analysis

Just like the adversaries can attack the localization schemes to make sensors' locations wrongly estimated, they can also attack the verification algorithm to make abnormal locations NOT to be detected by the VC. To achieve this goal, the attackers will compromise a sensor and force it to report fake neighborhood observation that is consistent with the claimed location. We illustrate such an attack in Figure 4.5. In the figure, sensor S_4 is compromised and localized at location L' that is far away from its true location L. If sensor S_4 reports the true observation $O_4 = (S_1, S_2, S_3)$, then the GFM algorithm will easily find inconsistencies because the estimated locations of sensors S_1, S_2 and S_3 are far away from location L'. To escape from being detected, S_4 may report a fake neighborhood observation $O_4 = (S_5, S_6, S_7)$, which

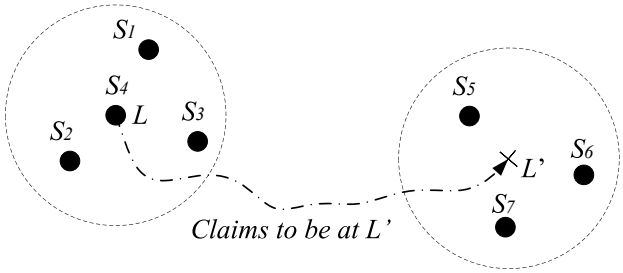


Figure 4.5 Attack to GFM algorithm

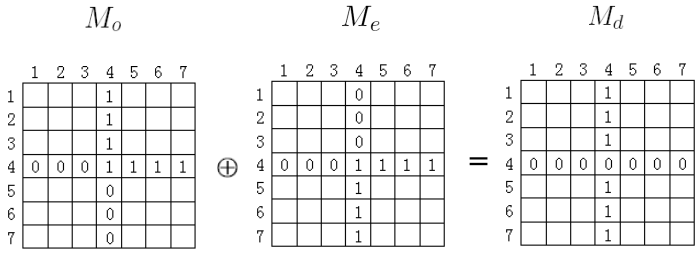


Figure 4.6 GFM matrixes under attacks

include sensors that are localized in the neighborhood of location L'.

We will use the above example to analyze GFM algorithm's performance. In Figure 4.6, the elements in the 4^{th} row and in the 4^{th} column of matrix M_d are shown. For simplicity, weight matrix M_w is not involved here. Based on this matrix, the metric values for sensor S_4 are $AD_4 = 0$, $PD_4 = 6$ and $AS_4 = 6$. The values of PD_4 and AS_4 are very high, hence, it is very probable that S_4 will be revoked.

To further mitigate the inconsistencies, S_4 may not broadcast its ID message, thus none of S_1, S_2 and S_3 can observe sensor S_4. The elements are recalculated in the difference matrix M_d, and the metric values become $AD_4 = 0$, $PD_4 = 3$ and $AS_4 = 3$. In this scenario, according to the definition of consistent neighbors in equation (4.10), the consistent-neighbor metric is $CN_4 = 0$, so that S_4 will be revoked during the final check at line (13)-(14) in Figure 4.4.

Inconsistency can be completed removed if sensors S_4, S_5, S_6 and S_7 are all compromised. Sensor S_4 does not broadcast its ID, and reports a fake neighborhood observation $O_4 = (S_5, S_6, S_7)$; meanwhile, sensors S_5, S_6 and S_7 all report to observe S_4. Such colluding attack is expensive to launch and is against our assumption in Section 2 that the majority of

sensors are benign in a local area. Therefore, we conclude that as long as the adversaries cannot compromise majority sensors in a local area, inconsistencies always exist and localization anomaly can be detected. Simulation results also prove the effectiveness and robustness of the GFM algorithm.

4.3.2 Greedy Filtering using Trustability-indicator

In GFT algorithm, VC computes a trustability indicator for each sensor and updates the indicator's value in multiple rounds. In each round, if a sensor's indicator is higher than the threshold, the sensor is accepted as correctly-localized sensor. Such iteration stops when all sensors' indicators become stable. Finally, the sensors that have indicator values lower than the threshold are detected and revoked.

4.3.2.1 Calculation of Trustability-indicator

The trustability indicator of sensor S_i ($i \in \{1, 2, \cdots, n\}$) is calculated in multiple rounds. Initially, all indicators are set to 0.5 in round 0. In round $k > 0$, sensor S_i's indicator is denoted by I_i^k and calculated by:

$$I_i^k = \frac{\sum_{S_j \in N_i} T_{ij}^k \cdot I_j^{k-1}}{\sum_{S_j \in N_i} I_j^{k-1}}, \tag{4.11}$$

where N_i denotes the set of sensors that can observe sensor S_i. For each sensor S_j in this set, the symbol I_j^{k-1} denotes S_j's indicator in the previous round. So actually I_i^k is calculated as a weighted average of all its neighbors' indicators in the previous round, where the weight T_{ij}^k is calculated based on the geographical relationship between S_i and S_j. In the following, we discuss how to calculate the weight factor in detail.

4.3.2.2 Calculation of Weights

According to the definition in equation (4.10), for each sensor that can observe S_i, if its estimated locations is within communication range of sensor S_i's estimated location, then this sensor is considered a consistent neighbor; otherwise, it is an inconsistent neighbor.

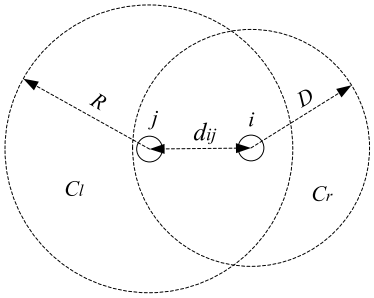


Figure 4.7 Compute temporary indicator

- **Compute the weight using consistent neighbor:** We discuss how to compute T_{ij}^k using a consistent neighbor. As shown in Figure 4.7, the distance between sensor S_i's and S_j's estimated locations is d_{ij}; R is the radius of the communication range; D is the anomaly degree. Verifying whether sensor S_i's localization error is smaller than D equals verifying whether S_i's true location is in circle C_r. We compute the following probability:

$$\begin{aligned} P(S_i \in C_r) &= P(S_i \in C_r | S_i \in C_l) \cdot P(S_i \in C_l) \\ &\quad + P(S_i \in C_r | S_i \notin C_l) \cdot P(S_i \notin C_l) \\ &\approx P(S_i \in C_r | S_i \in C_l) \cdot P(S_i \in C_l) \\ &= (S_o / S_l) \cdot f(d_{ij}), \end{aligned} \tag{4.12}$$

  where the approximation follows because the conditional probability $P(S_i \in C_r | S_i \notin C_l)$ is very small. S_o denotes the area of the overlapping region, and S_l denotes the area of the circle C_l. Function $f(x)$ is the segment-wise function defined in Figure 4.3(a). Because consistent neighbors help to enhance the trustability of sensor S_i, we increase S_i's previous indicator by the amount in equation (4.12), and get the weight T_{ij}^k as follows:

$$T_{ij}^k = I_i^{k-1} + P(S_i \in C_r) \tag{4.13}$$

- **Compute the weight using inconsistent neighbor:** We compute the probability that sensor S_i cannot be verified, namely, the probability that sensor S_i's true location is outside the circle C_r in Figure 4.7, such that:

$$P(i \notin C_r) = 1 - (S_o / S_l) \cdot f(d_{ij}) \tag{4.14}$$

To the opposite of consistent neighbors, inconsistent neighbors should reduce the trustability of sensor S_i's location, therefore, we decrease S_i's previous indicator by the amount in the above equation and calculate the weight as follows:

$$T_{ij}^k = \max\{I_i^{k-1} - P(S_i \notin C_r), 0\} \tag{4.15}$$

In each round, the VC calculates the weights using both consistent and inconsistent neighbors of a sensor, then update the sensor's indicator using equation (4.11).

4.3.2.3 Greedy Filtering Procedure

As shown in Figure 4.8, in each round, the VC updates each sensor's trustability indicator, then verifies any sensor if its indicator is greater than the threshold. If a sensor's indicator changes with negligibly small amount in two consecutive rounds, VC recognizes that the indicator has converged and stops updating its value. Threshold can be obtained through training on experimental data. We run GFT algorithm in non-attack environment and calculate the indicators for all sensors. Then we set the threshold according to the desired false alarm rate. If the false alarm rate is 0.5%, then the threshold will equal to the 99.5% percentile of all sensors' indicators.

Assign initial value of 0.5 to all sensors (1)
For round $k = 1$ to N (2)
  For each sensor S_i (3)
    update S_i's indicator from I_i^{k-1} to I_i^k (4)
    if $I_i^k >$ threshold (5)
      accept sensor S_i and stop updating its indicator (6)
    if $|I_i^k - I_i^{k-1}| < 0.05$ (7)
      stop updating the indicator in future rounds (8)
Verify sensors that have indicators greater than the threshold (9)

Figure 4.8 The GFT Algorithm

4.3.2.4 Security analysis

In GFT algorithm, consistent neighbors can increase a sensor's indicator, and inconsistent neighbors can decrease it. Knowing the filtering process of GFT, attackers will purposely avoid being revoked by generating as many consistent neighbors as possible for a victim sensor. To achieve this goal, the sensor being compromised needs to keep silent of its ID; otherwise, it will be observed by surrounding sensors that will become its inconsistent neighbors. On the other hand, the sensors that are around the estimated location of the compromised sensor need to claim to "observe" this sensor, so that they could become consistent neighbors of this sensor. However, we notice as long as majority sensors are benign in a local area, they will become inconsistent neighbors and reduce the sensor's trustability. Simulation results demonstrate that GFT algorithm is resilient and has satisfiable performance under such attacks.

4.4 In-region Location Verification

In this section, we propose a lightweight algorithm that the VC can use to perform in-region verification. Before we describe the algorithm, we first need to figure out how to determine the region inside which a sensor's location should be verified.

4.4.1 Verification Region Determination

Given a location-based application, we define the *verification region* as the physical region inside which the sensor should be verified *if and only if* the application goal can be achieved:

$$\text{Application goal is fulfilled} \Leftrightarrow L_i \in V_i, \tag{4.16}$$

where L_i is the true location of sensor S_i, and V_i is the verification region for sensor S_i. Notice that the verification region for different sensors may be different. In addition, we define two variants to the above region, and name them *sufficient* region and *necessary* region respectively:

$$\text{Application goal is fulfilled} \quad \Leftarrow \quad L_i \in \tilde{V}_i, \tag{4.17}$$

$$\text{Application goal is fulfilled} \quad \Rightarrow \quad L_i \in \hat{V}_i, \tag{4.18}$$

where $\tilde{V}_i$ is the sufficient region and $\hat{V}_i$ is the necessary region. From the geographical point of view, region $\tilde{V}_i$ is fully contained by region $\hat{V}_i$.

In the following, we use a location-based surveillance application to demonstrate how the verification region can be determined.

4.4.1.1 Battlefield surveillance

In battlefield surveillance, sensors that detect a suspicious object will inform the control center to destroy the object. There can be many approaches to determining the projection location, for simplicity, we assume it is the average location of all the estimated locations of the sensors that detect the tank. We do not require sensors to be equipped with any special hardware such as antennas or distance-measuring devices, however, the determination of the verification regions of sensors will be different depending on whether the angle and distance information are available.

Some notations we will use are as follows: the bombing range is C_x, the center and radius of C_x is P_x and R_x, the sensing distance is R_s, the true location of the tank is L_t, and sensor S_i's true and estimated location is L_i and L_i'.

- **Distance and angle are known.** If both distance and angle information are available, i.e., each sensor reports the distance and the direction that the object is from itself, then the problem is:

$$\text{Given } |\overrightarrow{L_iL_t}| = d_i, \angle\overrightarrow{L_iL_t} = \alpha_i, \text{ and } C_x,$$
$$\text{determine the verification region } V_i, \text{such that}$$
$$L_t \in C_x \Leftrightarrow L_i \in V_i. \tag{4.19}$$

  We use the example in Figure 4.9 to further clarify the above problem. In the figure, two sensors S_1 and S_2 detect the tank. The bombing center P_x and the bombing range C_x are given to the VC. The distance and angle from sensor S_1 to the tank are d_1 and α_1, respectively. The problem is to determine a verification region V_1 that satisfies equation (4.19). The solution to this problem is not difficult to find: We can move C_x for distance

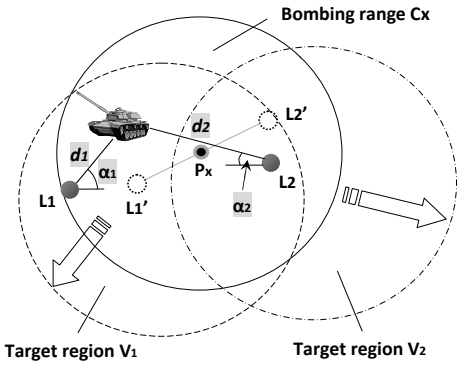


Figure 4.9 Verification regions (when distance and angle can be measured)

d_1 in the direction of $\alpha_1 + \pi$ to get V_1. It can be proved that if and only if sensor s_1 is inside V_1, then the tank will be inside C_x. The verification region V_2 for sensor s_2 can be determined similarly. The solution is formalized as followings:

$$V_i = T(C_x, \alpha_i + \pi, d_i), \tag{4.20}$$

where $T(x, y, z)$ is the operation of geometry translation, i.e., it moves every point of region x along the direction of y for a distance of z.

- **Angle is known and distance is unknown.** Since d_i is unknown, the solution given by equation (4.20) does not exist. Thus we find a pair of sufficient/necessary regions:

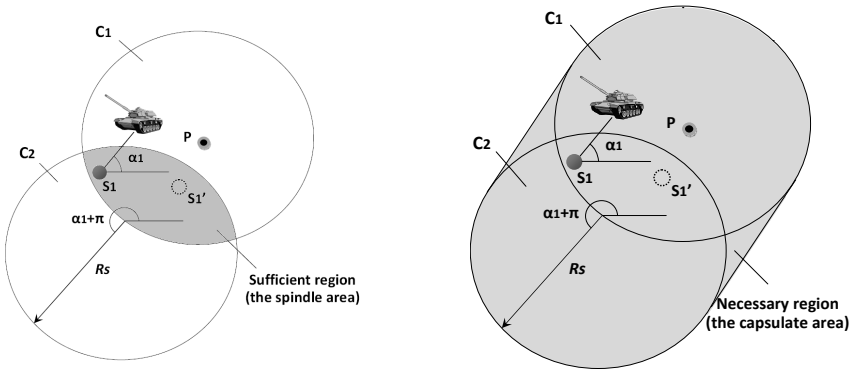
$$\text{Given } |\overrightarrow{L_i L_t}| \leq R_s, \angle\overrightarrow{L_i L_t} = \alpha_i, \text{ and } C_x,$$

$$\text{determine a pair of verification regions, such that}$$

$$L_t \in C_x \Leftarrow L_i \in \tilde{V}_i, \tag{4.21}$$

$$L_t \in C_x \Rightarrow L_i \in \hat{V}_i, \tag{4.22}$$

where the constraint $|\overrightarrow{L_i L_t}| \leq R_s$ means that the tank is within the sensing distance of the sensor, otherwise, it would not have been detected. The solution are not difficult to find: The sufficient region is the intersection of all transformed regions that have moving distance in the range $[0, R_s]$, and the necessary region is the union of all such transformed



(a) Sufficient region $\tilde{V_1}$ (b) Necessary region $\hat{V_1}$

Figure 4.10 Verification regions for S_1 (known angle, unknown distance)

regions, namely:

$$\tilde{V_i} = \bigcap_{d_i \in [0, R_s]} T(C_x, \alpha_i + \pi, d_i) \tag{4.23}$$

$$\hat{V_i} = \bigcup_{d_i \in [0, R_s]} T(C_x, \alpha_i + \pi, d_i) \tag{4.24}$$

where $\bigcap$ is the intersection operation and $\bigcup$ is the union operations. An example in Figure 4.10 shows how the sufficient region and necessary region can be calculated.

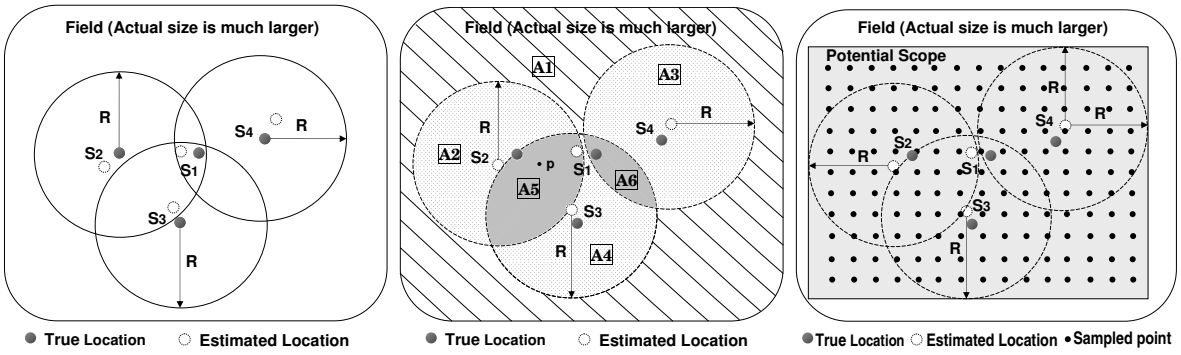
- **Angle is unknown and distance is known.** Without the information of the direction, the solution given by equation (4.20) does not exist either. In this scenario, the sufficient region should be a circle centered at the bombing center P_x and with a radius of $R_x - d_i$. It is easy to verify that if the sensor is in this circle, then the tank will be in a circle centered at P_x with radius R_x, which is exactly the bombing range C_x. Similarly, the necessary region can also be determined. The solutions are given by following formulas:

$$\tilde{V_i} = C(P_x, R_x - d_i), \tag{4.25}$$

$$\hat{V_i} = C(P_x, R_x + d_i), \tag{4.26}$$

where $C(x, y)$ denotes a circle centered at location x with radius y.

- **Both angle and distance are unknown.** If neither angle or distance information is known, then we can only infer that the tank is within the sensing distance R_s from the



(a) s_1's neighborhood (b) ECRs (c) Potential Scope

Figure 4.11 Snapshot of the field: sensor s_1 has three neighbors s_2, s_3, s_4

sensor. The sufficient region and necessary region are two concentric circles that center at P_x, and their radius are $R_x - R_s$ and $R_x + R_s$ respectively:

$$\tilde{V}_i = C(P_x, R_x - R_s), \tag{4.27}$$

$$\hat{V}_i = C(P_x, R_x + R_s) \tag{4.28}$$

We have analyzed how the verification region can be determined under different scenarios. In other application, the verification region may or may not be explicitly given, and it relies on the VC to derive it properly. Basically, the application goal should be quantified and the geographical relationship between sensors and the object of interest needs to be explored, then the verification region or a pair of sufficient/necessary regions can be derived using equation (4.16) or equations (4.17)(4.18).

4.4.2 In-region Verification

In this section, we propose a lightweight algorithm that the VC can use to perform in-region verifications. This algorithm also utilizes sensors' neighborhood observations as we described in section 4.1. Basically, if two sensors observe each other, then the VC consider them to be a pair of "confirmed" neighbors. Then, VC derives a probability distribution for each sensor, which indicates how probably the sensor is at each point in the field. The distribution function can be either continuous or discrete. In the continuous version, the in-region confidence is

t_1 \ t_2	0	1	2	3	4	5	6	7	8	9	10	•••
1	16.67	83.33	0	0	0	0	0	0	0	0	0	
2	2.17	22.24	75.59	0	0	0	0	0	0	0	0	
3	0.22	4.09	27.18	68.51	0	0	0	0	0	0	0	
4	0	0.58	6.09	30.52	62.81	0	0	0	0	0	0	
5	0	0.07	0.93	8.39	32.74	57.87	0	0	0	0	0	
6	0	0	0.16	1.51	10.25	36.21	51.87	0	0	0	0	
7	0	0	0.05	0.29	2.60	12.46	37.14	47.46	0	0	0	
8	0	0	0	0.05	0.40	3.20	14.80	38.28	43.27	0	0	
9	0	0	0	0	0.06	0.84	4.42	16.74	38.04	39.90	0	
10	0	0	0	0	0.02	0.15	0.96	5.58	19.11	37.99	36.19	
•••												

Row Index t_1: Number of confirmed neighbors of a sensor
Column Index t_2: score of the district where the sensor truly resides

Figure 4.12 Statistical results from simulation

computed by taking the integral of the distribution function within the verification region; In the discrete version, the in-region confidence is the sum of the probabilities of all points within the verification region. We discuss the technical details in following sections.

4.4.2.1 Scored districts

Notice that communication range is the region that centers at a sensor's *true* location with radius R. Here we define a variant named *estimated communication range (ECR)* which is a circle that centers at the sensor's *estimated* location. The VC uses the ECRs of a sensor's confirmed neighbors to divide the field into several regions. Each region has a score which is the number of the ECRs that cover this region. An example is shown in Figure 4.11(b), the solid and hollow circles represent sensors' true and estimated locations respectively. Sensor S_1 has three confirmed neighbors: sensors S_2, S_3 and S_4. The field is divided into six regions which belong to three districts. The 0-scored district contains region A_1; the 1-scored district contains regions A_2, A_3 and A_4; and the 2-scored district contains regions A_5 and A_6.

We notice that a sensor may not be inside the highest-scored district, because the ECRs are *estimated* communication ranges and may not cover a sensor's true location. However, we guess the probability that a sensor is inside a higher-scored district is higher, given that sensors are localized with reasonable errors. In the following, we will prove this conjecture using simulation data.

If the data can be collected from the field, then they can be directly used for our purpose.

Otherwise, simulations should be conducted using proper network parameters. In our simulation, 600 sensors are randomly deployed in a square field of $300m \times 300m$. The communication range is $R = 20m$. Each sensor averagely has 12 neighbors in its communication range. The environmental disturbance is quantified by $f = 10\%$, which means a sensor has 90% chance to receive a beacon message from its neighbor. For each sensor, we record the number of confirmed neighbors it has, then we divide the field into several scored districts and record which scored-district contain the sensor's true location.

The results of all sensors are summarized in Figure 4.12. In this table, row index is the number of confirmed neighbors, and column index is the score of district. Element $T(t_1, t_2) = p$ means that among all the sensors that have t_1 confirmed neighbors, $p\%$ of them are inside a district that scores t_2. For instance, $T(3, 2) = 27.18$ means among all the sensors that have 3 confirmed neighbors, 27.18% are inside score-2 district.

The VC uses the above training table to assign different *weights* to different districts. We still use the example in Figure 4.11(b) to explain the procedure. In the figure, there are three districts divided for sensor s_1, which have scores 0, 1 and 2 respectively. Since S_1 has 3 confirmed neighbors, so VC refers to the 3rd row in the table. The weights corresponding to score 0,1 and 2 are $T(3, 0) = 0.22$, $T(3, 1) = 4.09$ and $T(3, 2) = 27.18$. Based on these weights, the probabilities that sensor S_1 resides inside different districts can be computed. For example, the probability S_1 is inside score-2 district is $27.18/(0.22 + 4.09 + 27.18) = 0.8631$. The formula for calculating the in-district probabilities is given by:

$$P_r(L_i \in D_{im}) = \frac{T(n_i, m)}{\sum_{k \in M_{inc}} T(n_i, k)}, \forall m \in M_{inc}, \tag{4.29}$$

where D_{im} is the m-scored district, n_i is the number of s_i's confirmed neighbors, and M_{inc} is the set of distinct scores of all districts divided for sensor s_i.

4.4.2.2 Continuous distribution

The probability density function (pdf) specifies the probability density that a sensor may reside at different points in the field. By equation (4.29), we have calculated in-district probabilities. We adopt the assumption that the probability distribution within one district is

uniform. Because a sensor will have exactly the same number of confirmed neighbors at two points within one district, thus the two points cannot be statistically differentiated. Based on this assumption, we can compute the in-district probability density by dividing the in-district probability by the district's area. For instance, in Figure 4.11(b), if the area value of score-2 district is $12m^2$, and the probability that S_1 is in this district is 0.8631, then the probability density at any point within score-2 district is $0.8631/12 = 0.0719$. The formula for calculating the pdf function for sensor s_i can be given by:

$$pdf_i(l) = \frac{P_r(L_i \in D_{im})}{S(D_{im})}, \forall l \in D_{im}, \tag{4.30}$$

where the dividend $P_r(L_i \in D_{im})$ is the in-district probability in equation (4.29), and the divisor $S(D_{im})$ is the area value of the m-scored district D_{im}.

4.4.2.3 Discrete distribution

Since it is relatively expensive to calculate the district's area $S(D_{im})$ in equation (4.30). Therefore, we now discuss how to calculate a discrete distribution that does not involve complex computations.

We notice that the weights corresponding to the zero-scored district (the first column in the training table in Figure 4.12) have very small values, and secondly, the area of zero-scored district is very large (approximates the area of the whole field). Therefore, the probability density inside zero-scored district will be very small. Based on this observation, in our algorithm, VC determines a *potential scope* and only focus on the interior of this scope. We use the example in Figure 4.11(c) to explain the determination of the potential scope. In the figure, sensor S_1 has three confirmed neighbors S_2, S_3 and S_4. The boundaries of the potential scope are the tangent lines of the three ECRs. Obviously, the potential scope will cover all nonzero-scored districts.

Within the potential scope, the VC samples points uniformly and assigns different probability to each point. And the probabilities of all points in a district should sum to the in-district probability calculated by equation (4.29). Therefore, the probability mass function (pmf) of sensor s_i is given by:

$$pmf_i(l) = \begin{cases} \frac{P_r(L_i \in D_{im})}{N(D_{im})}, & l \in P_i, \\ 0, & \text{otherwise}, \end{cases}$$

where the dividend $P_r(L_i \in D_{im})$ is the in-district probability in equation (4.30), $N(D_{im})$ is the number of sampled points in district D_{im}, and P_i is the potential scope of sensor S_i.

4.4.2.4 Verification confidence

The verification confidence is the confidence that a sensor can be verified within the verification region. If the distribution is continuous, the in-region confidence is computed by taking the two-dimensional integral of the probability density function in equation (4.30) within the verification region:

$$P_r(L_i \in V_i) = \int\int_{V_i} pdf_i(x, y)dxdy \tag{4.31}$$

If the distribution is discrete, the in-region confidence is the addition of probabilities of all points in the verification region:

$$P_r(L_i \in V_i) = \sum_{l \in P_i} pmf_i(l) \cdot I(l \in V_i), \tag{4.32}$$

where I is an indicator function which outputs 1 if $l \in V_i$, and outputs 0 otherwise.

After the VC provides the verification confidence to the control center, the center will compare the confidence with an application-specific threshold to make proper decisions. In the battlefield surveillance application, sensor S_i detects a tank and its location is verified by the VC. Then the verification confidence is compared with a threshold to decide whether to project a bomb. Therefore,

$$\begin{cases} P_r(L_i \in V_i) \geq t & \Rightarrow \text{project a bomb}, \\ P_r(L_i \in V_i) < t & \Rightarrow \text{not project a bomb}, \end{cases}$$

where t is the threshold and its value is application specific. If t is set high, false alarms can be easily filtered out and bombs will not be wasted on non-existing targets; if t is set low, few targets will escape from being destroyed, but some bombs might be exploded for nothing. No matter how the threshold is determined, we expect a verification algorithm that is not sensitive to threshold values. Namely, when the thresholds fall into a large range of values, both the

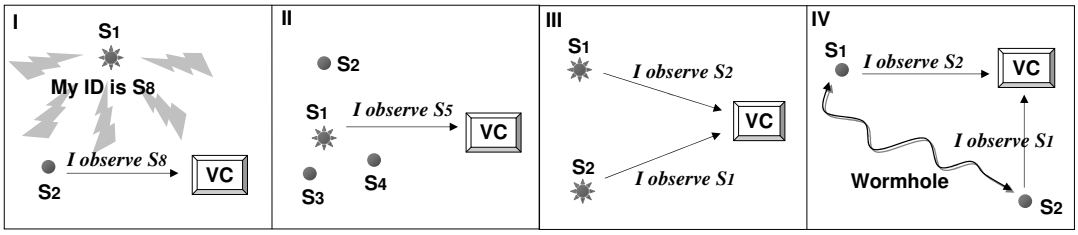


Figure 4.13 Attacks to the verification algorithm

false negative rate and false positive rate can maintain at satisfactory levels. In Section 6, we will study the false positive/negative rates of our verification algorithm, and it shows that our algorithm has good verification performance with various thresholds settings.

4.4.2.5 Security analysis

When WSNs are deployed in hostile environment, adversaries may purposely disrupt the verification process. Since the VC purely rely on sensors' neighborhood observations to derive probability distributions of sensors' locations, the attackers will try to generate misleading neighborhood observations. We illustrate the possible attacks in Figure 4.13 which contains four sub-figures: (I) a compromised sensor broadcasts an incorrect ID, hence, other sensors report wrong neighborhood observation; (II) a compromised sensor directly reports wrong neighborhood observation; (III) two sensors that are localized far away from each other cooperate and claim to observe each other; (IV) wormhole attackers record beacon messages at one location, tunnel them through a wired link and replay at another location. Thus, the sensors at the two ends of the wormhole will both report to observe the other.

In our algorithm, the VC confirms the neighboring relationship of two sensors only when they can observe each other. Since the first two attacks (attack I and II) generate asymmetric observations between sensors, their neighborhood observations will not be considered by the VC for further use. However, the colluding attack and the wormhole attacks (attack III and IV) are both capable to generate symmetric observations between a pair of sensors. Therefore, a far-away neighbor of a sensor may contribute a nonzero-scored district in the field. However, we notice only high-scored districts bear high densities and have major impact on the distribution

function, thus as long as the *majority* of a sensor's confirmed neighbors are benign, the high-scored districts will not be greatly distorted.

We introduce a parameter c to indicate the colluding degree. These neighbors provide fake location references but they collaboratively make their estimated locations close to each other. This attack is the most sophisticated attack the adversary can launch, and the cost is that they need to compromise many sensors to cheat on one sensor's verification. In the simulation, we set the value of c in the range of $(0, 50\%)$, and we notice that the distribution functions are not much distorted. In other words, as long as the colluding degree is minority, the verification confidence can be calculated with small errors, which still helps applications make correct decisions.

4.5 Simulation Study

4.5.1 Simulation Setup and Parameters

In the standard setup, we deploy $n = 600$ sensors randomly in a $300m \times 300m$ square field. The communication range is $R = 20m$. Each sensor can observe 12 neighbors on average. Sensors' localization errors follow the Gaussian distribution and the mean and deviation of the distribution are both $10m$. Among all sensors in the field, we randomly select $p\%$ sensors that are localized with errors greater than the anomaly degree D, while other sensors are localized with errors uniformly distribute within the range of $[0, D]$. We consider distance measuring errors and use the following equation: $\widetilde{d} = d + d * n_f$, where d is the distance between sensors' true locations and n_f is the *noise factor* that uniformly distributes within range $[-0.1, 0.1]$.

We consider the following attack model: first, the victim sensor does not broadcast its ID message, thus its true neighbors will not observe it; second, multiple sensors collude together and claim to observe the sensor. These colluding sensors all have been compromised and are physically near the claimed location of the victim. The percentage of compromised neighbors, namely the colluding degree, is denoted by c. We assume benign neighbors are the majority in a local area, so that c is in the range of (0, 0.5).

The criteria we use to evaluate the verification performance are Detection Rate (DR) and

False Positive Rate (FR). They are conflict criteria, because when thresholds are set low, both DR and FR will increase, and vice versa. A good verification algorithm should be designed to achieve high detection rate but low false positive rate. In the following, we will study both the on-spot and in-region verification performance of our verification algorithms.

4.5.2 On-spot Verification Performance

We formally define the evaluation criteria for the on-spot verification. The estimated locations with errors larger than D are called abnormal locations, while those with errors smaller than D are called correct locations. Therefore, Detection Rate (DR) is defined as the ratio between the number of abnormal locations being detected and the number of all abnormal locations, namely:

$$DR = N_{df}/N_f, \tag{4.33}$$

where N_{df} is the number of abnormal locations detected by VC, and N_f is the number of all abnormal locations.

Similarly, the false positive (false alarm) rate is defined by:

$$FP = N_{dt}/N_t, \tag{4.34}$$

where N_{dt} is the number of correct locations that are wrongly detected by VC, and N_t is the number of all correct locations.

In the following sections, we vary the value of anomaly degree, the percentage of abnormal locations, and the total number of sensors, so that we can test how the verification algorithms perform in different scenarios.

4.5.2.1 Impact of anomaly degrees

Figure 4.14 shows the Receive Operating Characteristic (ROC) curves for GFM and GFT algorithms under different values of D. Each point on the ROC curve is obtained by averaging on twenty sample runs in the field. Based on the curves, we can conclude that (1) Both verification algorithms achieve higher detection rate for higher anomaly degree. For example,

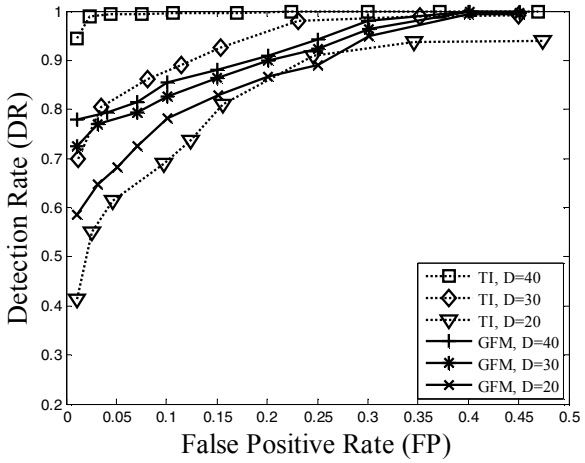


Figure 4.14 Impact of anomaly degrees

when anomaly degree is $20m$, $30m$ and $40m$ respectively, GFT's detection rates is 65%, 85% and almost 100%, given that the false positive rate is 5%. This is in accordance with our expectation, because the farther a sensor's estimated location is away from its true location, the more inconsistency will be introduced into the system. (2) Secondly, we can see that GFM outperforms GFT in detecting minor anomalies, but underperforms GFT in detecting severe anomalies. For example, when anomaly degree $D = 20m$, GFM's detection rate is 75% and GFT's is 70%, given false positive rate is 10%. However, when $D = 40m$, GFM's detection rate is 85% which is lower than GFT's detection rate 100%. Therefor, VC can select verification algorithm according to the given anomaly degree. If severe abnormal locations are more desired to be detected, GFT algorithm will be better, otherwise GFM will be the choice.

4.5.2.2 Impact of the percentage of abnormal locations

In this experiment, we vary the percentage p% of abnormal locations, in order to test whether GFM and GFT algorithms can still provide solid verifications.

We maintain the false positive rate at 5%, change the value of p and calculate the corresponding detection rate. The results are depicted in Figure 4.15. We can observe that the detection rate for both algorithms decreases as the percentage of abnormal locations increases. Moreover, both algorithms are more tolerant to large percentages when the anomaly degree is

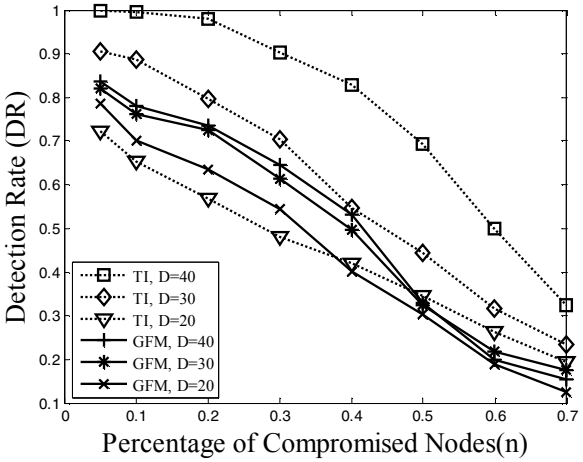


Figure 4.15 Impact of attack percentage

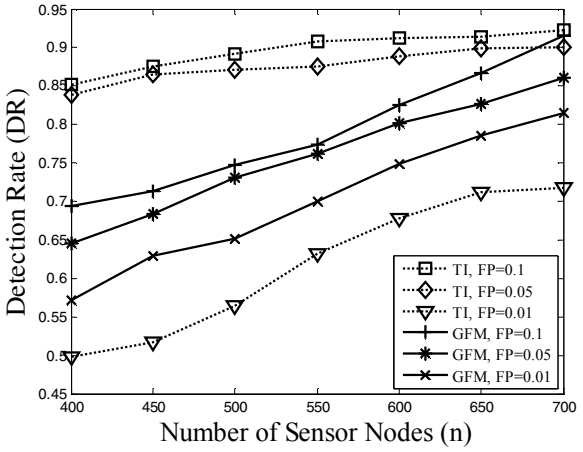


Figure 4.16 Impact of network density

higher. For example, when anomaly degree is $40m$ and the percentage is 30%, GFT's detection rate is above 90%. But when anomaly degree is $30m$ and the percentage is still 30%, GFT's detection rate drops to 70%.

From this experiment, we conclude that the GFM and GFT verification algorithms are more suitable for verifying sensors' location where the majority sensors are correctly localized, i.e., when over 90% sensors are correctly localized, then the abnormal locations will be easily detected.

4.5.2.3 Impact of network density

The goal of this experiment is to study how network density can impact the performance of the verification algorithms. In Figure 4.15, we observe that both GFM's and GFT's detection rates increase as the network density increases. For example, when the number of sensors increases from 400 to 700, the detection rate of GFM raises from 65% to 85%, given that the false positive is maintained at 5%. Secondly, we can observe that GFM outperforms GFT when the false positive rate is maintained at relatively small value, but GFM underperforms GFT when false positive rate increases. For instance, when $FP = 1\%$, the curve representing GFM's detection rate is always above the other curve that representing GFT's, and when $FP = 10\%$, the opposite happens. Therefore, the VC can choose the verification algorithm according to the application requirement about the false positive rate.

4.5.3 In-region Verification Performance

In this section, we test how the in-region verification performs in the battlefield surveillance application. We deploy N tanks randomly in the field. The sensing distance is $10m$, so that sensors within $10m$ from the tank can detect it. We set the bombing radius as $R_x = 25m$ and set the bombing center as the average location of estimated locations of all sensors that detect a same tank. We assume the anomaly degree is $D = 25m$. Among all sensors, 10% are localized with errors larger than $25m$ and the rest are localized with errors in the range $[0, 25m]$.

We define the evaluation criteria as follows:

$$DR = N_{df}/N_d, \tag{4.35}$$

$$FP = N_{dt}/N_t, \tag{4.36}$$

where N_{df} is the number of tanks that should be destroyed, N_d is the number of tanks that are indeed destroyed (based on the verification information), N_t is the number of tanks that should NOT be destroyed, and N_{dt} is the number of tanks that are wrongly destroyed.

We test the in-region verification performance in four different scenarios which we described

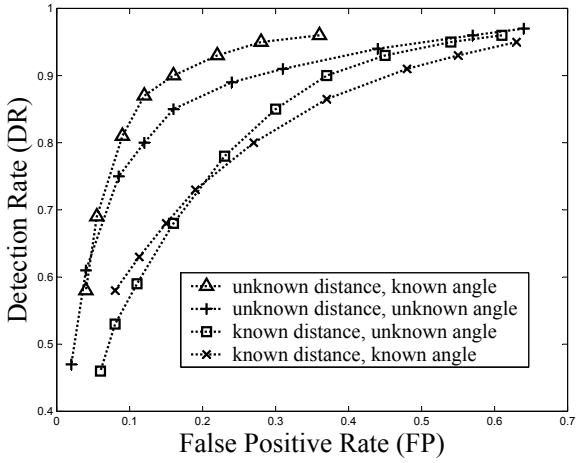


Figure 4.17 Missile projection

in section 5.1. The ROC curves are shown in Figure 4.17. We can observe that the best performance is achieved under the "unknown distance known angle" situation: when FR is 15%, DR can reach 90%. The curve obtained in the situation of "unknown distance unknown angle" performs the second: when FR is 15%, DR is about 85%. We also notice angle measurements can improve the verification performance, but only slightly. For example, the curve that corresponds to "known angle" is about 3% higher than than the peer which corresponds to "unknown angle".

Based on the simulation results, we can conclude that the proposed in-region verification algorithm can be effectively applied to battlefield surveillance application. Even when no distance or angle can be measured by sensors, the algorithm yields satisfactory (or even better) verification results.

4.6 Conclusion

We notice that detection and removal of information anomaly is very important for constructing a trustworthy communication system. In this work, we studied location anomalies detection and proposed several algorithms, which can also be extended to detect general communication anomalies. More precisely, we proposed three lightweight verification algorithms. The on-spot verification algorithms, including GFM and GFT algorithm, verify whether the

locations claimed by sensors are far from their true spots beyond a certain distance. Sensors' neighborhood observations are explored for the information inconsistencies. In-region verification verifies whether a sensor is inside an application-specific verification region. We proposed a probabilistic method that computes the confidence that a sensor is inside the verification region. Our work takes the first step tempting to integrate the application requirements into determining the trustability of sensors' estimated locations. Moreover, compared to previous works, our proposed verification algorithms are more lightweight, effective and robust. They do not require any dedicated or expensive infrastructures in the field; they yield satisfactory verification results to a variety of applications, which is approved by the simulation results; furthermore, they are resilient to malicious attacks and can be used in hostile environments.

CHAPTER 5. DISTRIBUTING CONFIDENTIAL LOCATION INFORMATION USING NETWORK CODING

5.1 Introduction

Network coding [3][67] is a new message forwarding technique that allows a forwarder to encode multiple input messages together to form an output one. Unlike the traditional approach that always duplicates every forwarding message, network coding is able to maximize the throughput of multicast networks. In 2003, Li et al. [67] further proved that linear network coding is sufficient to achieve the optimal throughput, which is the minimum of the max-flows from the single source to sinks. Because of this nice property, network coding has been widely used not only in wired networks [21][39], but also in wireless networks [9][19][56][57][79][86].

Network coding systems suffer from various malicious attacks, including wiretapping attacks (passive attacks) and pollution attacks (active attacks). While great research efforts have been devoted to prevent or mitigate pollution attacks, there have been relatively less promising solutions that efficiently and effectively prevent or mitigate wiretapping attacks.

Wiretapping attack was defined by Cai et al. in [14]. In this attack, adversaries are capable to wiretap or eavesdrop on a subset of the links of some network coding system and gain access to the information transmitted through the links. The problem is how to prevent information from leaking to adversaries. Traditional approaches require end-to-end or hop-by-hop encryption. However, for the end-to-end approach, if a large amount of the encrypted information is obtained by adversaries, they will conduct known-ciphertext attack to learn the real contents of the messages. For the hop-by-hop encryption, since every pair of neighboring nodes need to share a secret key, not only extra communication overhead will be caused by the key setup/update process, but also much delay will be introduced by encryption/decryption

at every hop.

To defend against wiretapping attacks without encryption, an approach is to introduce randomness into the source messages. Several solutions [8][14][33][53] based on this idea have been proposed. These solutions can be categorized into two classes: Shannon-secure ones and weakly-secure ones. The difference is that Shannon-secure ones do not allow any information leakage, while weakly-secure ones do not allow any *meaningful* information leakage. As an example, given two information symbols x_1 and x_2, in weakly-secure schemes, the adversaries may learn the value of $x_1 + x_2$ but not x_1 or x_2 alone; in Shannon-secure schemes, they cannot learn neither of x_1, x_2 nor the combination $x_1 + x_2$.

All existing secure solutions for network coding consume more communication overhead compared to insecure codings. First, when random symbols are introduced into the message vector sent from the source, the real information symbols that can be transmitted are decreased, i.e., the multicast capacity is decreased. If we want to multicast the same amount of information as that in insecure coding system, then more multicast session will be involved and thus more network energy will be consumed. Second, secure solutions enlarge the size of the finite field over which the coding is done (for the purpose of maintaining security properties). Therefore, more binary bits are needed to represent symbols and encoding coefficients, which increase the bandwidth consumptions. In resource-constrained networks such as wireless sensor networks, bandwidth is very limited and network energy is a very scarce resource, the wireless communications between nodes can drain their energy quickly, which will shorten the network lifetime and severely impact network functionalities.

In this research, we aim to design efficient weakly-secure network coding schemes that are suitable for energy-constrained networks. We focus on weak security because some network coding applications may not have perfect secure requirements in practice. Fortunately, if the security requirement can be weakened, not only the loss in multicast rate but also the enlargement of finite field can be avoided, which are the two major drawbacks associated with previous solutions.

We propose two coding schemes against wiretapping attacks. The first is a basic scheme

which introduces only one random symbol into the messages vector sent by the source, and increases the finite field to the extent such that the last element in the vector can be protected. The second is an advanced scheme which introduces no random symbol. It maintains the maximum multicast capacity and retains the size of the finite field. Both our schemes utilize a special permutation functions to generate randomness. The input and output range of the function are the finite field of the coding scheme, and the function can be implemented using Linear Feedback Shift Register (LFSR) or its variants.

5.2 Problem Statement

5.2.1 System Model

A network coding system can be represented by a tuple $(\mathcal{G}, \alpha, \mathcal{U})$. In the tuple, $\mathcal{G} = (V, E)$ is a directed acyclic graph, and V and E are the set of nodes and edges of $\mathcal{G}$. The capacity of each edge indicates the maximum average rate of information that can be transmitted on this edge. The single source α generates and sends out a message vector X every time unit, which consists of n symbols in a finite field F_q, i.e., $X = (x_1, x_2, \cdots, x_n)^T \in F_q^n$. A set of users denoted by $\mathcal{U}$ receive and recover the multicast information, and $|\mathcal{U}| = u$. In linear coding systems, the message on any outgoing edge of a node is a linear combination of the messages on its incoming edges. So the message on any edge can be written as a linear combination of the source symbols. We denote the message transmitted on edge e by $\mathcal{T}_e X$, where $\mathcal{T}_e$, named the *encoding vector*, is a row vector over finite field F_q^n.

5.2.2 Threat Model

In wiretapping attack, the adversaries are able to wiretap or eavesdrop on a subset of the edges in a network coding system and gain access to the information transmitted on these edges. Precisely, they can wiretap on one (but no more than one) subsect in a collection $\mathcal{A} = \{A_1, A_2, \cdots, A_{|\mathcal{A}|}\}$, where A_i represents a set of edges. We can define an eavesdropping matrix M_i of dimension $k_i \times n$ corresponding to each A_i, where each row in M_i represents an encoding vector and k_i is the maximum number of linearly independent encoding vectors on

edges in set A_i. For each M_i, the information available to the eavesdropper is a set of linear equations denoted by $M_iX = B_i$, where B_i is a vector consisted of k_i symbols heard on the k_i edges in A_i.

Besides the above model, a second model has also been used by researchers to quantify the capacity of the eavesdroppers. In this model, the adversary can access to at most k edges. This model can be viewed as a special case of the first model when the collection $\mathcal{A}$ contains all non-empty subsets of at most k edges in the network.

In this research, we consider the most powerful wiretapping model, where the wiretapper can obtain any subset of at most $n-1$ edges in the network. This model is equivalent to the second model when k is equal to $n-1$. Note that we do not need to consider any adversary more powerful than what we consider here, because if a wiretapper can obtain n or more linearly independent equations, then he is essentially of no difference from normal users in terms of information amount, thus there would be no solution to prevent information leakage.

5.2.3 Security Goals

The security goal is to prevent the source information from leaking to the adversaries.

We can categorize the security goals into Shannon-secure and weakly-secure. The difference between them is that Shannon-secure does not allow the leakage of any information about the source, while weakly-secure does not allow the leakage of any *meaningful* information. Formally speaking, for any $M_i \in \mathcal{M}$, Shannon security requires that $H(X|M_iX = B_i) = H(X)$, while weak security requires that $H(x_i|M_iX = B_i) = H(x_i)$, for all x_i in message vector X. As an example, given two source symbols x_1 and x_2, weak security allows the adversaries to gain the value of $x_1 + x_2$, but Shannon security does not permit such leakage.

In this research, we focus on designing weakly-secure network coding schemes. We notice weakly-secure coding can be used in some applications that do not have very strict secure requirements. It has already drawn some research interests [8][53] in recent years. Fortunately, if the security requirement can be weakened to weakly-secure, not only the loss in multicast rate but also the enlargement of finite field over which the source symbols and encoding coefficients

are selected, can be avoided. This feature is very desirable in some coding system (such as wireless sensor networks) where the energy is a scarce resource and communication overhead needs to be minimized.

5.3 Scheme I: A Basic Scheme

In this section, we propose a scheme that can be applied to an existing insecure coding and make it weakly-secure. All linear operations including addition, substraction, multiplication and division are conducted over finite field F_q.

5.3.1 Randomizing at the Source

Assume the maximum multicast capacity is n in a coding system, namely, the source can multicast a vector consisted of n symbols to all the receivers in one time unit. In our scheme, the source firstly selects a random symbol w from the finite field F_q and inserts it to the end of the message vector. So the message vector sent by the source is $X = (x_1, x_2, \cdots, x_{n-1}, w)^T$, where x_1 to x_{n-1} are information symbols and w is the inserted random symbol.

The source utilizes a random permutation function h together with w to randomize the information symbols. The input and output of function h are both elements in the finite field F_q. This function serves as a random number generator and can be constructed using Linear Feedback Shift Register (LFSR) [72] or its variants [22][36]. As we will discuss later, LFSR is very efficient for low-cost nodes to implement in hardware. Note that the construction of h function is public, so all the nodes and even adversaries know how to build it.

The source uses the h function and computes a vector $X' = (x'_1, x'_2, \cdots, x'_n)^T$ where

$$\begin{cases} x'_1 = x_1 + h(w) \\ x'_2 = x_2 + h^2(w) \\ \cdots\cdots \\ x'_{n-1} = x_{n-1} + h^{n-1}(w) \\ x'_n = w \end{cases}$$

Here $h^2(w) = h(h(w))$ means that h function is applied twice to parameter w; $h^3(w)$ means h function is applied for three times to parameter w, and so forth.

After the source calculates the vector X', it multiplies a full rank matrix C of dimension $n \times n$ to X' and get $X'' = CX'$. The full-rank matrix C should have the property that the last row of its reverse matrix C^{-1} is not in the linear span of all eavesdropping matrixes that have dimension $(n-1) \times n$. We will explain why we require such property for matrix C in Section III-*D*. Finally, the source injects vector X'' into the network, and the vector can be multicast to all receivers using any feasible insecure network code.

5.3.2 Decoding at the Receivers

Given that the existing network code is feasible (though it is not secure), each destination node first decodes each elements in vector X'' by solving a set of linear equations. Then the receiver obtains vector X' by $X' = C^{-1}X''$, where C^{-1} is the reverse of full-rank matrix C. Now since the receiver knows the value of w, which is the last element in vector X', he can calculate x_1 to x_{n-1} by:

$$x_i = x'_i - h^i(w), i \in \{1, \cdots, n-1\}.$$

Therefore, the receiver can decode all information symbols.

5.3.3 Implementation Details

In our scheme, the permutation function can be implemented using the basic Linear Feedback Shift Register (LFSR) [72], or its variance such as the Shrinking Generator [22] and the Alternating Step Generator (ASG) [36].

LFSR is a simple circuit that consists of shift registers and XOR gates. For an L-bit LFSR, L shift registers are used to store one bit each, and the circuit is constructed based on an irreducible polynomial over the finite field F_2 with a degree of L. For example, 16-bit Galois LFSR will be constructed according to the feedback polynomial $x^{16}+x^{14}+x^{13}+x^{11}+1$. Tables such as [95] have been created to show where the taps should be placed. The sequence that an LFSR produces has good statistical properties, and can be considered as a series of pseudo

random numbers [72]. Being pseudo random, and not random, comes from the fact that the sequence does follow a mathematical formula. However, if the seed of the LFSR is kept secret, the LFSR can be used to create L-bit random secrets. Besides the basic LFSR, the Shrinking Generator and the ASG can also be used and they provide better statistical properties in term of period length and linear complexity.

5.3.4 Security Analysis

We notice that in order to solve for an information symbol x_i, both the symbol $x'_i = x_i + h^i(w)$ and the random number w should be known, otherwise, the randomness introduced by the h function cannot be removed.

In order to get the value of w, the eavesdropper will try to perform linear transmission to the set of equations he has obtained and solves for the last variable x'_n. In other words, he will try to use $MX'' = MCX' = B$, where M denotes any eavesdropping matrix of dimension $(n-1) \times n$, to obtain $I_n X' = x'_n$, where $I_n = (0, \cdots, 1)$ is a row vector whose last element is 1 and other elements are 0. To achieve the transformation from MC to I_n, the adversary needs to find a row vector t, such that $tMC = I_n$, or equivalently, the adversary needs to find a row vector t such that $tM = I_n C^{-1}$. However, since the last row of matrix C^{-1} is not in the linear span of matrix M according to the rule when we construct C, such a vector t does not exist and the above transmission cannot be possibly conducted. Therefore, the wiretapper is unable to remove the randomness and cannot solve for any x_i. The proposed network code scheme is weakly-secure.

5.3.5 Discussion

Compared to insecure network code, our basic secure scheme achieves weakly secure but gives up some performance. First, the multicast capacity is reduced by $1/n$ because in one time unit, only $n-1$ information symbols can be multicast to all receivers instead of n.

Second, since we require the last row of the reverse matrix C^{-1} is not in the linear span of any eavesdropping matrix M, the size of the finite field has to be enlarged. Precisely, when

we construct matrix C, we first select the last row in matrix C^{-1}, then complete other rows in C^{-1} while ensuring all rows are linearly independent such that the matrix is full rank. Finally, we reverse C^{-1} back to obtain C. In this process, finite field is enlarged in the first step. Since the last row of C^{-1} cannot be in the linear span of any eavesdropping matrix M, the finite field should be large enough so that such a row can be found, namely, $q^n > \begin{pmatrix} |E| \\ n-1 \end{pmatrix} q^{n-1}$, where the scaler at the right side is the total number of eavesdropping matrix that has dimension $(n-1) \times n$. Therefore, the lower bound is $q > \begin{pmatrix} |E| \\ n-1 \end{pmatrix}$. We notice this lower bound is larger than that required by insecure schemes, i.e., $q > u$, especially in large networks which consist a large number of edges. The direct consequence is that we need more binary bits to represent symbols and encoding coefficients, so more communication bandwidth will be consumed. Consider an example where $u = 20$ users exist in a network which contains $|E| = 100$ edges and the optimal multicast rate is $n = 10$. The bandwidth consumption is $q > 20$ using insecure codes, but is increased to about $q > 10^{18}$ using the secure one.

To improve the basic scheme, we propose an advanced scheme in the next section. The nice features of the advanced scheme include: first, no random symbols need to be inserted into the message vector, thus the maximum multicast capacity is maintained. Second, both the information symbols and encoding coefficients can be taken from the same finite field as in insecure schemes. Third, neither the source nor the destination nodes need to make transmissions to the message vector, i.e., no full-rank matrix C is needed to translate X' into X'', thus computation overhead is reduced.

A comparison between our basic and advanced schemes and some previous solutions is shown in TABLE 5.1.

5.4 Scheme II: An Advanced Scheme

5.4.1 Randomizing at the Source

In the advanced scheme, no random symbols are inserted into the message vector. Instead, the source applies the h function to the information symbols directly. The h function is

constructed in the same way as in the basic scheme, i.e., it randomly maps an integer in F_q into another integer in F_q.

If the multicast capacity of the network is 2, then the message vector sent by the source is $X' = (x'_1, x'_2)^T$ where

$$\begin{cases} x'_1 = x_1 + h(x_2) \\ x'_2 = h(x'_1) + x_2 \end{cases}$$

In general case where the multicast capacity of the network is n ($n \geq 2$), then the message vector $X' = (x_1, \cdots, x'_n)^T$ is:

$$\begin{cases} x'_1 = x_1 + h(x_2) \\ x'_2 = x_2 + h(x_3) \\ \cdots\cdots \\ x'_{n-1} = x_{n-1} + h(x_n) \\ x'_n = h(x'_1) + \cdots + h(x'_{n-1}) + x_n \end{cases}$$

Then the source can multicast X' to all receivers using any feasible insecure network code.

5.4.2 Decoding at the Receivers

After a destination node receives a set of linear equations, it first solves for the elements in X', then it calculates $x_n, x_{n-1}, \cdots, x_1$ iteratively using the following formulas:

$$\begin{cases} x_n = x'_n - h(x'_1) + \cdots + h(x'_{n-1}) \\ x_{n-1} = x'_{n-1} - h(x_n) \\ x_{n-2} = x'_{n-2} - h(x_{n-1}) \\ \cdots\cdots \\ x_1 = x'_1 - h(x_2) \end{cases}$$

In this way, the receivers decode all information symbols.

5.4.3 Security Analysis

In this subsection, we will prove that if the wiretappers cannot obtain more than $n-1$ linearly independent equations, then the proposed advanced coding scheme is weakly-secure.

Table 5.1 Performance Comparison (Eavesdropping capacity k=n-1)

	Security	**Multicast Rate**	**Finite Field Size**
Insecure Schemes	None	n	u
Cai et al. [14]	Shannon	1	$\binom{\lvert E\rvert}{\text{n-1}}$
Feldman et al. [33]	Shannon	<1	$< \binom{\lvert E\rvert}{\text{n-1}}$
Bhattad et al. [8]	Weakly	n	$\binom{\lvert E\rvert}{\text{n-1}}+1$
Our Basic	Weakly	n-1	$\binom{\lvert E\rvert}{\text{n-1}}$
Our Advanced	Weakly	n	u

5.4.3.1 Simplified Scenario

We start by proving the security property in a simplified situation where the length of message vector is two, i.e., $X' = (x'_1, x'_2)^T$. The wiretapper can wiretap an arbitrary edge e in the network, and obtain one linear combination of the symbols, namely,

$$\mathcal{T}_e X' = t_1 \cdot x'_1 + t_2 \cdot x'_2 = b, \tag{5.1}$$

where $\mathcal{T}_e = (t_1, t_2)$ is the encoding vector on edge e, and b is the symbol transmitted on edge e. We analyze the security property through the following three cases depending on whether t_1 or t_2 is zero.

- Case 1: $t_1 = 0$, $t_2 \neq 0$, so equation (5.1) is written as:

$$t_2 \cdot x'_2 = t_2 \cdot [h(x_1 + h(x_2)) + x_2] = b \tag{5.2}$$

  We introduce a free variable y which can take arbitrary value in F_q, then the solutions to the above equation can be expressed in terms of y, such that:

$$\begin{cases} x_1 = y - h(b/t_2 - h(y)) \\ x_2 = b/t_2 - h(y) \end{cases}$$

  Given any value for y within finite field F_q, a pair of solutions for (x_1, x_2) can be calculated using the above equations. In other words, x_1 and x_2 can take any value over F_q according to different value of y in the field. The conditional entropy $H(x_i|\mathcal{T}_e X' = b)$ is equal to $H(x_i)$. This network code is weakly-secure.

- Case 2: $t_1 \neq 0$, $t_2 = 0$, then equation (5.1) is written as:

$$t_1 \cdot x'_1 = t_1 \cdot (x_1 + h(x_2)) = b. \tag{5.3}$$

Here x_2 can be viewed as a free variable which can take arbitrary value in F_q. For a given value of x_2, the symbol x_1 is calculated by $x_1 = b/t_1 - h(x_2)$. Since both x_1 and x_2 can take arbitrary value in F_q, i.e., $H(x_i|\mathcal{T}_e X' = b) = H(x_i)$ for any information symbol $x_i \in X$, our scheme is weakly-secure.

- Case 3: If $t_1 \neq 0$ and $t_2 \neq 0$, then equation (5.1) is equivalent to the following:

$$t_1 x_1 + t_1 h(x_2) + t_2 h(x_1 + h(x_2)) + t_2 x_2 = b. \tag{5.4}$$

Similar to Case 1, we introduce variable y and express the solutions for x_1 and x_2 in terms of y, such that:

$$\begin{cases} x_1 = y - h((b - t_1 y - t_2 h(y))/t_2) \\ x_2 = (b - t_1 y - t_2 h(y))/t_2 \end{cases}$$

Given an arbitrary value for y in the finite field F_q, the solutions for x_1 and x_2 can be calculated accordingly. Again we have $H(x_i|\mathcal{T}_e X' = b) = H(x_i)$ for x_1 and x_2, which indicates security property of our scheme.

5.4.3.2 General Scenario

Now we analyze the general scenario where the multicast capacity is n ($n \geq 2$).

Since the wiretapper can obtain at most $n-1$ linearly independent equations, which means that he cannot solve for all the elements in vector X', so he cannot use equations (5.1) to find out the values of any symbol from x_1 to x_n. However, he may try to perform linear transmissions to the equation set, and we will analyze such attacks in details at the following.

By taking linear transformation of the set of equations $MX' = B$, where M denote any eavesdropping matrix of dimension $(n-1) \times n$, the wiretapper can obtain a function of $x'_1, \cdots, x'_n$, which has the following general form:

$$t_1 x'_1 + \cdots + t_n x'_n = d, \tag{5.5}$$

where $t_1, \cdots, t_n$ are coefficients over the finite field F_q. Since we don't specify the edges (and the encoding vectors on these edges) being eavesdropped, t_i can take arbitrary values. Equation (5.5) can also be written as:

$$t_1 \cdot [x_1 + h(x_2)] + \cdots + t_{n-1} \cdot [x_{n-1} + h(x_n)] +$$
$$t_n \cdot [h(x_1 + h(x_2)) + \cdots + h(x_{n-1} + h(x_n)) + x_n] = d \tag{5.6}$$

Now we analyze the security property in two cases.

- Case 1: If $t_n \neq 0$, we introduce $n-1$ variables $y_i \in F_q$, for $i = 1, \cdots, n-1$. For any given value of the vector $(y_1, \cdots, y_{n-1})^T \in F_q^{n-1}$, we can give the following solutions for $x_1, \cdots, x_n$ that satisfy equation (5.6):

$$\begin{cases} x_n = \frac{d - \sum_{i=1}^{n-1} t_i \cdot y_i}{t_n} - \sum_{i=1}^{n-1} h(y_i) \\ x_{n-1} = y_{n-1} - h(x_n) \\ \cdots \\ x_2 = y_2 - h(x_3) \\ x_1 = y_1 - h(x_2) \end{cases}$$

  The correctness of above solutions can be easily verified. Due to space limitation, we omit the proof here. Note that in this solution, $x_{n-1}, \cdots, x_1$ can be iteratively expressed in term of the free variables $y_1, \cdots, y_{n-1}$. To eliminate the randomness caused by the free variables, the adversary may manipulate coefficients t_i. If all t_i for $i = 1, \cdots, n-1$ are zero, then the linear random part $\sum_{i=1}^{n-1} t_i y_i$ becomes zero and is eliminated. However, there is no way to get rid of the non-linear part $\sum_{i=1}^{n-1} h(y_i)$, so the randomness in x_n cannot be removed. Since other symbols are iteratively expressed in terms of x_n, their values cannot be solved either.

- Case 2: when $t_n = 0$, then there will be at least one coefficient among t_1 to t_{n-1} that is non-zero. We denote these non-zero coefficients by $\{t_{k_1}, \cdots, t_{k_m}\} \subseteq \{t_1, \cdots, t_n\}$, where $1 \leq k_1 < \cdots < k_m \leq n-1$. It can be proved that any x_i where $i \notin \{t_{k_1}, \cdots, t_{k_m}\}$ is free variable, and they can take any value in F_q without affecting the correctness of the equation. Now we introduce a set of k_m free variables $z_1, z_2, \cdots, z_{k_m}$ over F_q. Then we

are able to express information symbols $x_{k_1}, \cdots, x_{k_m}$ in term of free variables only, such that:

$$\begin{cases} x_{k_m} = \frac{d - \sum_{i=k_1}^{k_m} t_i \cdot z_i}{t_{k_m}} - h(x_{k_m+1}) \\ x_{k_{m-1}} = z_{k_{m-1}} - h(x_{k_{m-1}+1}) \\ \cdots \\ x_{k_1} = z_{k_1} - h(x_{k_1+1}) \end{cases}$$

The correctness of above solutions can be verified. Again, we see that none of the information symbols can be solved and the entropy $H(x_i|MX' = B) = H(x_i)$ for all $x_i \in X$. Hence, this advanced network code scheme achieves weak security.

5.5 Conclusion

In this research, we propose efficient schemes for securing linear network coding against wiretapping attacks. They exploit the permutation function and the interleaved relationship between transmitted symbols to achieve randomness. The advanced scheme maintains the multicast capacity as that can be achieved in insecure network codings, and does not enlarge the finite field over which the coding should be done. Therefore, it is especially suitable for resource-constrained wireless networks. To the best of our knowledge, our advanced scheme is the first solution to wiretapping attacks that have the above two desired properties. In the future, we will evaluate our schemes by implementing it on MicaZ nodes.

CHAPTER 6. SUMMERY

6.1 Conclusion

In this research, we focus on designing a secure location-aware communication system for energy-constrained wireless network such as wireless sensor networks. There are three key aspects when building such a system, namely, location determination, location anomaly detection, and location information distribution. For each of the three aspect, we design many practical techniques to defend against a variety of malicious attacks. Particularly, we study the following important problems: (1) providing correct location estimations for sensors in presence of wormhole attacks and pollution attacks, (2) detecting location anomalies according to the application-specific requirements of the verification accuracy, and (3) preventing information leakage when using network coding for multicasting location information. To solve each problem, we propose several innovative algorithms that are efficient for wireless sensor networks in terms of hardware cost, computation overhead and communication overhead.

In summary, the contribution of our research are as follows:

- We first study the problem of providing reliable location information in presence of wormhole attacks. We propose a dynamic anchor-regrouping localization scheme. Compared with others, our scheme does not require dedicated hardware such as high-precision measuring device to be quipped on sensor nodes. Moreover, our scheme is more energy-efficient than other schemes since no extra communication overhead are incurred on sensors.

- We also address the problem of error propagation or pollution attacks to the localization in WSNs. We proposed a robust localization scheme which is based on the novel

notion of confidence tag to quantify the accuracy of sensor's location estimation. To our knowledge, our scheme is the first one to address the problem of the proliferation of wrong locations. Simulation results demonstrate the effectiveness of our schemes in enhancing the reliability (availability and accuracy) of location information for wireless sensor networks.

- We study the problem of detecting and revoking location anomalies. We proposed two schemes to verify whether the location claimed by a sensor deviates from the true one more than a certain distance. Compared to previous works, our verification algorithms are very lightweight, because they do not rely on verifiers or any prearranged infrastructures in the sensor field.

- We also study the in-region verification problem which is to determine whether a sensor is inside an application-specific area. We proposed a probabilistic algorithm where the verification center can calculate the confidence that a sensor can be in-region verified. Our research takes the first step tempting to integrate the application requirements into determining the trustability of sensors' locations.

- We propose two secure coding schemes which can effectively prevent information leakage to eavesdroppers. Using our schemes, the wiretappers can at most obtain linear combinations of information symbols, but cannot solve for any single symbol. Compared other schemes, our advanced scheme have two noticeable benefits for energy-constrained networks: First, it does not introduce extra communicating into the network and thus achieves maximum multicast capacity. Second, it does not enlarge the size of the finite field from which the coding coefficients are selected. Therefore, our scheme consumes less communication resource than previous works and are very suitable for energy-constrained wireless networks.

6.2 Future Work

To provide reliable location estimations for sensors in WSNs, we propose several algorithms to defend against wormhole attacks and pollution attacks. However, within this research area, there exist many more challenges and there are still a lot of space of improvements.

- Wormhole attack is a very detrimental and easy-to-launch attack. It can be classified as anchor-to-sensor wormhole and sensor-to-sensor wormhole depending on the victims around the end-points of the wormhole tunnel. The formal attack will mislead sensors to refer to beacon messages sent form far-away anchors, while the latter attack will make far-apart sensors to believe into residing in the same neighborhood area. As a result, the network topology based on the neighboring relationships will be disrupted and sensors' locations cannot be correctly estimated. Our proposed dynamic regrouping scheme effectively prevents the anchor-to-sensor wormhole attack. However, preventing sensor-to-sensor wormhole attack is still an unresolved area and deserves future research efforts. Especially for multihop localization schemes where not all sensor nods can receive anchors' beacon messages, and thus need to use other sensors' references, the sensor-to-sensor wormhole will directly impact their localization performance.

- We propose that anchors broadcast with a regrouping range to form local groups for defending against wormhole attacks. Since current technology enables anchors to adjust their transmission power within several levels, it will be interesting to investigate how dynamically changed regrouping range can impact wormhole detection rate. Actually, current localization schemes do not consider the coordinating opportunities between sensors and anchors, and much potential improvement may be introduced when involving anchors' self-adaptive behaviors, such as adjusting transmitting power levels based on sensors' feedbacks. Such adaptive/coorparative localization is especially beneficial for mobile WSNs where localization needs to be performed on a regular basis.

- For defending against pollution attack, we propose two efficient algorithms for calculating a confidence tag for each sensor's location. However, the calculation is based on the

assumption of using trilateration method when a sensor localizes itself. There are many other localizing techniques which are also subjected to pollution attack. For example, in DV-hop schemes where sensors utilize the number of hops to anchors, if adversaries compromise a node and send out wrong hop counts, then all down-steam nodes will be influenced. We will further investigate this problem and keep on improving our solution.

For detecting location anomalies in wireless sensor network, we study both the on-spot verification which is to determine whether a sensor is exactly localized at its claimed location, and the in-region verification which is to verify if a sensor resides within an application-specific region. In the future, we will continue the research in this area with the following goals.

- First, our approaches have the assumption of a centralized Verification Center (VC) which performs the verification task and reports to the base station or control center. However, when sensor network is composed of thousands of nodes, we need to design distributed solutions to tackle with the scalability issue properly, such as cluster based verification or hierarchical VC structure.

- For the in-region verification, we use the military surveillance application as an example for demonstrating the determination of verification region. In the future, we will investigate other location-based applications such as target tracking and geographical routing, and study how the verification regions can be tuned according to the application functionalities and network parameters.

- Our research takes the first step toward the lightweight location verification system. In the future, we are planning to set up testbed and apply our schemes to real-life WSNs, for testing and evaluating the performance and limitations of our approaches.

In this research, we propose efficient schemes for securing linear network coding against wiretapping attacks. They exploit the permutation function and the interleaved relationship between transmitted symbols to achieve randomness. In the future, we will evaluate our schemes by implementing it on sensor node hardware. Moreover, although many algorithms have been proposed for secure network coding, some problems in this topic are still unresolved.

- Besides weakly security which is to protect each single information symbol multicast by the source, in some security-sensitive applications, it is important to conceal any combinations of the information and achieve shannon secure. Previous schemes either sacrifice the multicast capacity and/or enlarge the size of finite field on which the coding is performed, which reduces the benefits that network coding can bring to multicast communications. In addition, when network coding schemes are applied to real wireless sensor networks, there are many practical issues such as link delays and cyclic graphs. It is challenging and interesting to study how to improve the network coding performance in such situations.

- Besides wiretapping attack, the adversaries can also compromise sensors directly and prevent the sinks from recovering source messages, which we call active attacks. For instance, they can launch DOS attack by selectively dropping messages or generating new messages that are not linearly independent of previous ones, and the result is that the number of linearly independent messages received by the sinks are decreased and information symbols cannot be resolved. The adversaries may also insert false sources into network coding systems, which violates the security goal of authenticity. It is challenging problem to design homomorphic signature scheme to achieve authenticity and integrity in linear coding (especially XOR coding) systems.

- To achieve secure and efficient multicast in energy-constrained networks, it is important to design scalable multicast architecture. We have done some research over the IP network, and we will continue the research of constructing multicast architecture in WSNs. Unless IP Network, there is no Antonymous System in WSNs and multicast cannot be explicitly categorized at intra-domain and inter-domain levels. We need to properly decompose WSNs into hierarchical levels based on which the source-encoding can be performed according to the membership within each level. Also, since sensors are very limited in energy supplies, the Multicast Agent (MA) for each "domain" cannot be fixed on a single sensor; in other words, the multicast architecture should dynamically load balance according to the energy consumption conditions of all sensor nodes.

BIBLIOGRAPHY

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks*, Vol. 38, No. 4, pp. 393–422, 2002.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102–114, 2002.

[3] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network Information Flow", in *IEEE Transactions on Information Theory*, Vol 46, No. 4, pp. 1204-1216, 2000.

[4] P. Bahl, and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System", in *IEEE INFOCOM*, 2000.

[5] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems", in *Proc. 22nd Annual International Cryptology Conference on Advances in Cryptology*, LNCS, Vol. 2442, pp. 354-368, 2002.

[6] M. Bellare, J. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures", in *Proc. Advances in Cryptology (EUROCRYPT'98)*, LNCS, Vol. 1403, pp. 236-250, 1998.

[7] P. Bergamo, and G. Mazzini, "Localization in Sensor Networks with Fading and Mobility", in *IEEE PIMRC*, 2002.

[8] K. Bhattad, and K. Narayanan, "Weakly Secure Network Coding", in *Proc. 1st Workshop on Network Coding, Theory, and Applications (NetCod)*, 2005.

[9] S. Biswas, and R. Morris, "ExOR: Opportunistic MultiHop Routing for Wireless Networks" in *Proc. ACM SIGCOMM*, pp. 133-143, 2005.

[10] S. Brands, and D. Chaum, "Distance-bounding protocols", in *Theory and Application of Cryptographic Techniques, pages 344C359*, 1993.

[11] S. Brands, and D. Chaum, "Distance-bounding protocols", in Workshop on the theory and application of cryptographic techniques on Advances in cryptology. Springer-Verlag New York, Inc., 1994, pp. 344C359.

[12] R. Brown, and P. Hwang, "Introduction to Signals and Applied Kalman Filtering", 1997.

[13] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices", in *IEEE Personal Communications*, Vol. 7, No. 5, pp. 284, 2000.

[14] N. Cai, and R. Yeung, "Secure Network Coding", in *Proc. International Symposium on Information Theory (ISIT)*, 2002.

[15] S. Capkun, and J. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor networks", in *IEEE INFOCOM*, 2005.

[16] S. Capkun, M. Cagalj, and M. Srivastava, "Secure Localization With Hidden and Mobile Base Statioins", in *IEEE INFOCOM*, 2006.

[17] W. Chen, J. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks", in *IEEE ICNP*, 2003.

[18] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", in Proc. IEEE Symposium on Security and Privacy, pp. 197-213, 2003.

[19] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing", in *Proc. ACM SIGCOMM*, pp. 169-180, 2007.

[20] D. Charles, K. Jian, and K. Lauter, "Signature for Network Coding", Technique Report MSR-TR-2005-159, Microsoft, 2005.

[21] P. Chou, Y. Wu, and K. Jain, "Practical Network Coding", in *Proc. Allerton Conference on Communication, Control, and Computing*, 2003.

[22] D. Coppersmith, H. Krawczyk and Y. Mansour, "The Shrinking Generator", in *Advances in Cryptology-CRYPTO '93*(LNCS 773), 22-39, 1994.

[23] J. R. Douceur, "The Sybil Attack", in *First International Workshop on Peer-to-Peer Systems (IPTPS)*, March 2002.

[24] A. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes", in *Proc. IEEE/ACM IPSN*, pp. 111-117, 2005.

[25] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "pairwise key pre-distribution scheme for wireless sensor networks", in Proc. ACM CCS, pp. 42-51, 2003.

[26] W. Du, L. Fang, and P. Ning, "LAD: Localization anomaly detection for wireless sensor networks", in Proceedings of IEEE International Parallel and Distributed Processing Symposium (*IPDPS*), 2005.

[27] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "key management scheme for wireless sensor networks using deployment knowledge", in Proc. IEEE INFOCOM, 2004.

[28] L. Doherty, K. S. Pister, and L. Ghaoui. "Convex position estimation in wireless sensor networks", in *IEEE INFOCOM*, 2001.

[29] E. Ekici, J. McNair, and D. Al-Abri, "A Probabilistic Approach to Location Verification in Wireless Sensor Networks", in Proceedings of IEEE International Conference on Communications (*ICC*), 2006.

[30] L. Eschenauer, and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", in *Proc. ACM CCS*, pp. 41-47, 2002.

[31] T. Eren, D. Goldenberg, W. Whiteley, Y. Yang, A. Morse, B. Anderson, P. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization", in *IEEE INFOCOM*, 2004.

[32] L. Fang, W. Du, and P. Ning, "A Beacon-Less Location Discovery Scheme for Wireless Sensor Networks", in *IEEE INFOCOM*, 2005.

[33] J. Feldman, T. Malkin, C. Stein, and R. Servedio, "On the Capacity of Secure Network Coding", in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, 2004.

[34] C. L. Fok, G. C. Roman, and C. Lu, "Mobile Agent Middleware for Sensor Networks: An Application Case Study," in *IPSN*, 2005.

[35] C. Fragouli, J. Boudec, and J. Widmer, "Network Coding: An Instant Primer", in *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 1, 2006.

[36] C.G. G*ü*nther, "Alternating step generators controlled by de Bruijn sequences", in *Advances in Cryptology - Eurocrypt '87* (LNCS 304), pages 5-14, 1988.

[37] W. Greene, "Econometric Analsis", third edition, Prince Hall, 1997.

[38] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate Pairing", in *Proc. 5th Algorithmic Number Theory Symposium (ANTS V)*, LNCS, Vol. 2369, pp. 324-337, 2002.

[39] C. Gkantsidis, and P. Rodriguez, "Network Coding for Large Scale File Distribution", in *Proc. IEEE INFOCOM*, pp. 2235-2245, 2005.

[40] C. Gkantsidis, and P. Rodriguez, "Cooperative Security for Network Coding File Distribution", in *Proc. IEEE INFOCOM*, 2006.

[41] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs", in *Proc. 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pp.119-129, 2004.

[42] IEEE 802.15 Working Group for Wireless Personal Area Networks (WPAN), http://ieee802.org/15/.

[43] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application", in *ACM Mobicom*, 1999.

[44] M. A. Hamid, M-O. Rashid, and C. S. Hong, "Routing Security in Sensor Network: Hello Flood Attack and Defense", in *the First International Conference on Next-Generation Wireless Systems (ICNEWS)*, January 2006, Dhaka.

[45] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Network", in *ACM MobiCom* 2003.

[46] T. Ho, R. Koetter, M. Médard, R. Karger, and M. Effros, "The Benefits of Coding over Routing in a Randomized Setting" in *Proc. International Symposium on Information Theory (ISIT)*, 2003.

[47] T. Ho, B. Leong, R. Koetter, M. Méard, M. Effros, and D. Karger, "Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding", in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2004.

[48] L. Hu, and D. Evans, "Localization for Mobile Sensor Networks", in *MobiCom*, 2004.

[49] L. Hu, and D. Evans, "Using directional antennas to prevent wormhole attacks", in *Network and Distributed System Security Symposium (NDSS)*, 2004.

[50] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks", in *IEEE INFOCOM*, 2003.

[51] X. Ji, and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling", in*IEEE INFOCOM*, 2004.

[52] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient Network Coding in the Presence of Byzantine Adversaries", in *Proc. IEEE INFOCOM*, pp. 616-624, 2007.

[53] K. Jain, "Security Based on network topology against the wiretapping attack", *IEEE Wireles Comm.*, 68-71, Feb 2004.

[54] B. Karp, and H. T. Kung, "Greedy Perimeter Stateless Routing", in *MobiCom*, 2000.

[55] Y. Ko, and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Adhoc Networks", in *ACM Mobicom*, 1998.

[56] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in The Air: Practical Wireless Network Coding", in *Proc. ACM SIGCOMM*, pp. 243-254, 2006.

[57] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding", in *Proc. ACM SIGCOMM*, pp. 397-408, 2007.

[58] S. Katti, D. Katabi, H. Balakrishnan, and M. Médard, "Symbol-level Network Coding for Wireless Mesh Networks", in *Proc. ACM SIGCOMM*, pp. 401-412, 2008.

[59] M. Krohn, M. Freeman, and D. Mazieres, "On-the-Fly Verification of Rateless Erase Codes for Efficient Content Distribution", in *Proc. IEEE Symposium on Security and Privacy*, pp. 226-240, 2004.

[60] K. Langendoen, and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison", in *Computer Networks: The International Journal of Computer and Telecommunications Networking*, v.43 n.4, p.499-518, 2003.

[61] L. Lazos, and R. Poovendran, "SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks", in *ACM Workshop on Wireless Security (WiSe)*, 2004.

[62] L. Lazos, R. Poovendran, and S. Capkun, "Rope: Robust Position Estimation in Wireless Sensor networks", in *ACM/IEEE IPSN*, 2005.

[63] A. Liu, P. Kampanakis, and P. Ning, TinyECC: Elliptic Curve Cryptography for Sensor Networks, http://discovery.csc.ncsu.edu/software/TinyECC/.

[64] D. Liu, P. Ning, and W. Du, "Attack-Resistant Location Estimation in Sensor Networks", in *Proceedings of The Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, 2005.

[65] D. Liu, P. Ning, and W. Du, "Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks", in *proceeding of International Conference on Disributed Computing Systems (ICDCS)*, 2005.

[66] D. Liu, and P. Ning, "Improving key pre-distribution with deployment knowledge in static sensor networks", ACM Transactions on Sensor Networks (ToSN), Vol. 1, No. 2, pp. 204-239, 2005.

[67] S. Li, R. Yeung, and N. Cai, "Linear Network Coding", in *IEEE Transactions on Information Theory*, Vol 49, No. 2, pp. 371381, 2003.

[68] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust Statistical Methods for Securing Wireless Localization in Sensor Networks", in *ACM/IEEE IPSN*, 2005.

[69] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements", in Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (*ACM SenSys*), 2004.

[70] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing Elliptic Curve Logorithms to Logorithms in a Finite Field", in *IEEE Transactions on Information Theory*, Vol 39, No. 5, pp. 1639-1646, 1993.

[71] V. Miller, "Short Programs for Functions over Curve", unpublished manuscript, *http://crypto.stanford.edu/miller/miller.pdf*, 1986.

[72] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, "Handbook of Applied Cryptography", CRC Press, ISBN: 0-8493-8523-7, October 1996.

[73] MIRACL, Multiprecision Integer and Rational Arithmetic C/C++ Library, http://www.shamus.ie/, Shamus Software Ltd.

[74] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network", in *ACM/IEEE IPSN* 2003.

[75] D. Nicolescu, and B. Nath, "Ad-Hoc Positioning Systems (APS)", in *IEEE GLOBECOM* 2001.

[76] D. Niculescu, and B. Nath, "Ad hoc positioning system (APS) using AoA", in *IEEE INFOCOM*, 2003.

[77] D. Niculescu, and B. Nath, "Dv based positioning in ad hoc networks", in Journal of Telecommunication Systems, 2003.

[78] C. E. Perkins, "Ad Hoc Networking", Addison-Wesley, Boston MA, 2001.

[79] D. Petrovic, K. Ramchandran, and J. Rabaey, "Overcoming Untuned Radios in Wireless Networks with Network Coding", in *IEEE Transactions on Information Theory*, Vol 52, No. 6, pp. 2649-2657, 2006.

[80] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system", in *ACM MobiCom*, 2000.

[81] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estin, "Sympath for the sensor network debugger", in *the 3rd ACM Conference on Embedded Networked Sensor Systems (Sensys)*, 2005.

[82] P. Rousseeuw, and A. Leroy, "Robust regression and outlier detection", Wiley-Interscience, 2003.

[83] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors", in *ACM MobiCom*, 2001.

[84] A. Savvides, H. Park, and M. B. Srivastava, "The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems", in *ACM WSNA*, 2002.

[85] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims", in ACM Workshop on Wireless Security *(WiSe)*,2003.

[86] S. Sengupta, S. Rayanchu, and S. Banerjee, "An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing", in *Proc. IEEE INFOCOM*, pp. 1028-1036, 2007.

[87] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking Moving Devices with the Cricket Location System", in *ACM MobiSys*, 2004.

[88] A. So, and Y. Yu, "Theory of Semidefinite Programming for Sensor Network Localization", in *ACM-SIAM Symposium on Discrete Algorithms (SODA)* 2005.

[89] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from Mere Connectivity", in *ACM MobiHoc*, 2003.

[90] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Improved MDS-Based Localization", in *IEEE INFOCOM*, 2004.

[91] B. Waters, and E. Felten, "Secure, Private Proofs of Location", Technical Report: Princeton Computer Science TR-667-03, 2003.

[92] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System", in *ACM Transactions on Information Systems*, vol. 10, No. 1, pp. 91-102, 1992.

[93] H. Wang, and Q. Li, "Efficient Implementation of Public Key Cryptosystems on MicaZ and TelosB Motes", Technical Report WM-CS-2006, College of William and Mary, 2006.

[94] D. Wang, D. Silva, and F. Kschischang, "Constricting the Adversary: A Broadcast Transformation for Network Coding", in *Proc. 45th Annual Allerton Conference on Communication, Control and Computing*, 2007.

[95] R. Ward, and T. Molteno, "Table of Linear Feedback Shift Registers", *http://www.physics.otago.ac.nz/px/research/electronics/ papers/technical-reports/lfsr_table.pdf*, technical Reports, University of Otago, New Zealand.

[96] Y. Wei, Z. Yu, and Y. Guan, "Location Verification Algorithms for Wireless Sensor Networks", in Proceedings of International Conference on Distributed Computing Systems *(ICDCS)*, 2007.

[97] Y. Wei, Z. Yu, and Y. Guan, "COTA: A Robust Multi-hop Localization Scheme in Wireless Sensor Networks", in *Proceedings of IEEE/ACM International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2006.

[98] Y. Wei, Z. Yu, and Y. Guan, "Efficient Weakly-Secure Network Coding Schemes against Wiretapping Attacks", in *Proceedings of the IEEE International Symposium on Network Coding (NetCod 2010)*, Toronto, Canada, June 2010.

[99] Y. Wei, Z. Yu, and Y. Guan, "A Novel Architecture for Secure and Scalable Multicast over IP Network", in *Proceedings of the 5th International ICST Conference on Security and Privacy in Communication Networks (SecureComm 2009)*, Athens, Greece, September 2009.

[100] Y. Wei, Z. Yu, and Y. Guan, "Localization Security in Wireless Sensor networks". Handbook of Research on Wireless Security, Y. Zhang (ed.), Idea Group Reference, April 2007.

[101] Y. Guan, Y. Wei, and Z. Yu, "DAR: A Lightweight Dynamic Anchor Regrouping Scheme for Securing Localization Service against Wormhole Attacks in Wireless Sensor Networks", in *Proceedings of the 6th International Conference on Networked Sensing Systems (INSS 2009)*, Pittsburgh, June 2009.

[102] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An Efficient Scheme for Securing XOR Network Coding against Pollution Attacks", in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2009)*, Brazil, April 2009.

[103] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan. "An Efficient Signature-based Scheme for Securing Network Coding against Pollution Attacks", in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM 2008)*, Phoenix, April 2008.

[104] Z. Yu, Y. Wei, and Y. Guan, "Key Management for Wireless Sensor Networks". Handbook of Wireless Mesh and Sensor Networking, G. Aggelou (ed.), McGraw-Hill International, February 2007.

[105] S. Zhang, S. Liew, and P. Lam, "Hot Topic: Physical-Layer Network Coding", in *Proc. MobiCom*, pp. 358-365, 2006.

[106] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for Content Distribution with Network Coding", in *Proc. International Symposium on Information Theory (ISIT)*, pp. 556-560, 2007.